COVALIDATION OF DISSIMILARLY STRUCTURED MODELS

DISSERTATION

Samuel A. Wright, Major, USAF

AFIT/DS/ENS/00-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/DS/ENS/00-02

COVALIDATION OF DISSIMILARLY STRUCTURED MODELS

DISSERTATION

Presented to the Faculty of the Graduate School of Engineering and Management
of the Air Force Institute of Technology
Air University in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Samuel A. Wright B.S., M.S.
Major, USAF

March 2001

Approved for public release, distribution unlimited

AFIT/DS/ENS/00-02

COVALIDATION OF DISSIMILARLY STRUCTURED MODELS

Samuel A. Wright, B.S., M.S
Major, USAF

Approved: Date:

19 DEC 00
Kenneth W. Bauer, Jr. (Chairman)

Mark E. Oxley 19 Dec 00
Mark E. Oxley

James T. Moore 19 Dec 00
James T. Moore

M.E. Franke 19 Dec 00
Milton E. Franke (Dean's Representative)

Accepted:

Robert A. Calico, Jr.
Dean, Graduate School of Engineering and Management

Preface

The problem of what to do with multiple models of the same system is one of some significance in these times of prolific modeling due to a) the untestable nature of many of today's large real-world systems and b) the increasing ease and cost-effectiveness of creating mathematical models. Here I provide a method of dealing with a system's multiple models, not only to provide some semblance of validation, but also to improve each model's performance based on information provided by the other models.

I would like to thank Dr. Ken Bauer for chairing the committee in this effort. His insight into the problem and style of direction were vital to the accomplishment of this research. Dr. Mark Oxley deserves my deep appreciation for his help in laying the theoretical foundation upon which this method stands. Many thanks also go to Dr. Jim Moore for providing guidance from a big picture as well as an editor's perspective. I can't imagine a committee that would better complement each other's strengths. Thanks also go to the many folks at AMCSAF for providing not only the application on which this methodology is tested, but also their keen insights into MASS and NRMO outputs that I couldn't have had otherwise.

Mostly, I thank my wonderful wife Vicki and amazing children Samantha, Sammy, Jacob, and Tabitha for providing knowledge of the truly important; wisdom and perspective that otherwise would have gone unlearned. Last (and first), I thank God, the provider of more than I'll ever comprehend. His divine guidance has proven invaluable.

Samuel A. Wright

Table of Contents

Page

List of Figures

Page

List of Tables

AFIT/DS/ENS/00-02

Abstract

A methodology is presented which allows comparison between models constructed under different modeling paradigms. Consider two models that exist to study different aspects of the same system, namely Air Mobility Command's strategic airlift system. One model simulates a fleet of aircraft moving a given combination of cargo and passengers from an onload point to an offload point. The second model is a linear program that optimizes aircraft and route selection given cargo and passenger requirements in order to minimize late- and non-deliveries. Further, the optimization model represents a more aggregated view of the airlift system than does the simulation. The two models do not have immediately comparable input or output structures, which complicates comparisons between the two models. I develop a methodology to structure this comparison and use it to compare the two large-scale models described above.

Models that compare favorably using this methodology are deemed covalid. Models that perform similarly under approximately the same input conditions are considered covalid in a narrow sense. Models that are covalid (in this narrow sense) may hold the potential to be used in an iterative fashion to improve the input (and thus, the output) of one another. I prove that, under certain regularity conditions, this method of output/input crossflow converges, and if the convergence is to a valid representation of the real-world system, the models are considered covalid in a wide sense. Further, if one of the models has been independently validated (in the traditional meaning), then a validation by association of the other model may be effected through this process. ,

COVALIDATION OF DISSIMILARLY STRUCTURED MODELS

I. Introduction

General

Mathematical models provide representations of real-world systems. These representations may take many forms: simulation, optimization, and regression, to name a few. Each requires an input set, employs some set of rules and relations, and returns a set of dependent variable(s) as output. Inputs consist of input variables that can be observed directly in the real-world system (such as the amount of a resource), and input parameters that cannot be directly observed (such as a service rate or an efficiency factor). It is possible that a real-world system has more than one model that may be employed to characterize some unique aspect of the system. For instance, a system that consists of implementing a schedule of events of variable duration could be modeled as a simulation in order to determine the (distribution of) total time required to complete the schedule. Such a model is descriptive in nature and could be used to forecast resource or personnel requirements. Alternately, an optimization model may be created in order to determine the shortest length of time in which the schedule may be implemented. This model optimizes some aspect of the system and could be used to study policy changes, such as improvement of the schedule driving the simulation model. Either model (or both) may be valid representations of the real-world system in question, and both, assuming their validity, should give results appropriate for their respective purposes.

In order for models to be useful, they should approximate the real-world system to the degree to which they are designed, a degree that varies with each system and its associated models. Model validation is defined as “substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with the study objectives” (Balci 1994). There is a myriad of techniques available to assess the validity of simulation models (Balci 1994, Sargent 1996b, or Sargent 1996a), and many of these techniques may be applied to more general classes of models. These techniques include both objective and subjective tests that may assess the validity of a model’s assumptions, structure, execution, or output performance. Depending on the study, different aspects of model validation may be of paramount importance. Here I focus on models’ output performances.

There is a large body of literature related to determining the output validity of a model for which real-world data can be obtained (see Law and Kelton 1991, p. 314-322 or Balci 1994 for examples). However, it is often the case that the system being represented cannot be sampled in order to make comparisons. In such a case, a method of model output validation is not as obvious. Sargent recommends the comparison of simulations to other validated simulations, as well as the comparison of simple simulations to known analytic results in cases of non-observable systems (Sargent 1996b). The use of similarly structured models (specifically, simulations) also has been proposed to assist in establishing model credibility (Diener, Hicks, and Long 1992). Kleijnen (1995) suggests that if relevant data are unattainable, a sensitivity analysis may be performed with the results compared against a system expert’s judgment.

Definitions and General Modeling Paradigm

In this research, I propose a method of *covalidation* in which models, either similarly (e.g. simulation and simulation) or dissimilarly structured (e.g. optimization and simulation), representing the same real-world system (perhaps for which output data are unattainable), may be compared. I refer to the *covalidation* of two (or more) models of similar or dissimilar structure representing the same real-world system. In general, *covalidation* is the process of comparing these models, mindful of each model's domain of applicability, with the object of relative substantiation. *Covalidity* can be thought of as a matter of degree. In a narrow sense, covalid models are models that perform similarly under approximately the same input conditions. In a wider sense, covalid models are models that can also be accepted as valid representations of the same real-world system. Further, if one of the models has been independently validated from the perspective of its intended purpose, models that are covalid relative to that model may be considered *valid by association.*

The process of determining the degree of covalidity of two or more models represents a new paradigm for situations in which two or more models are created for the same real-world system. This paradigm is explained using illustrations. Figure 1 represents the most basic of these illustrations but encapsulates the heart of my intent. As shown: 1) Models are created. 2) An attempt is made to use the outputs of the models to improve the inputs of the other models (a method I call "output/input crossflow"). 3) The models are compared one to another to determine their closeness to each other and/or reality (I assess the outputs as well as the gradients of the outputs and term the method "gradient analysis"). A poor comparison leads the modeler to modify the models and return to the output/input crossflow.

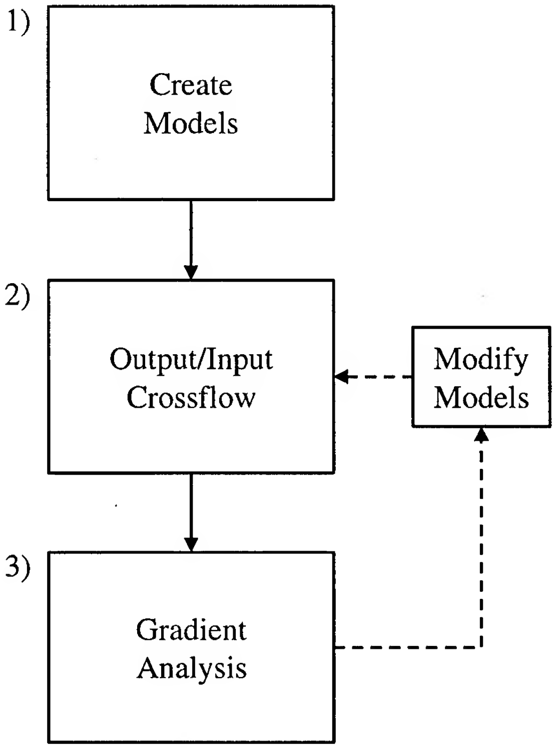

Figure 1: Modeling Paradigm

Figure 2 expands upon the modeling paradigm illustrated in Figure 1. Figure 2 breaks out the major steps of Figure 1 into more specific tasks that help define the methodology. I summarize this methodology in the following steps in Figure 2: 1a) Candidate models are built to represent the same real-world system. 2a) Strengths of each model are exploited through an informational crossflow. This crossflow utilizes knowledge and insights cultivated from the performance of each model to improve the performance of the other model(s) in an iterative scheme. 2b) The result of the crossflow is a convergence of input and output values (for each model) to a fixed-point that is considered to represent the intended modeling situation most closely. 3a) An evaluation of the relative closeness of the respective fixed-points occurs. This closeness represents the degree of communality of function that exists amongst the models. If the degree of communality is low, some or all of the models are not representing the same situation and require modification (and returning to Step 2a above). 3b) A determination is made concerning the covalidity of the models.

Relative closeness between models indicates narrow-sense covalidity. Mutual closeness to a standard and/or to the real-world system (if such a comparison is possible) indicates wide-sense covalidity. If models are covalid in the narrow sense, but not the wide sense, the models do not adequately represent the real-world system, and all require modification (and returning to Step 2a above). Further, it may be the case that no real-world data exists, yet one of the models has been validated by means other than output comparison, or has been simply accepted by acclamation (accredited). Models that are narrow-sense covalid with such a model may be considered valid (accredited) by association.

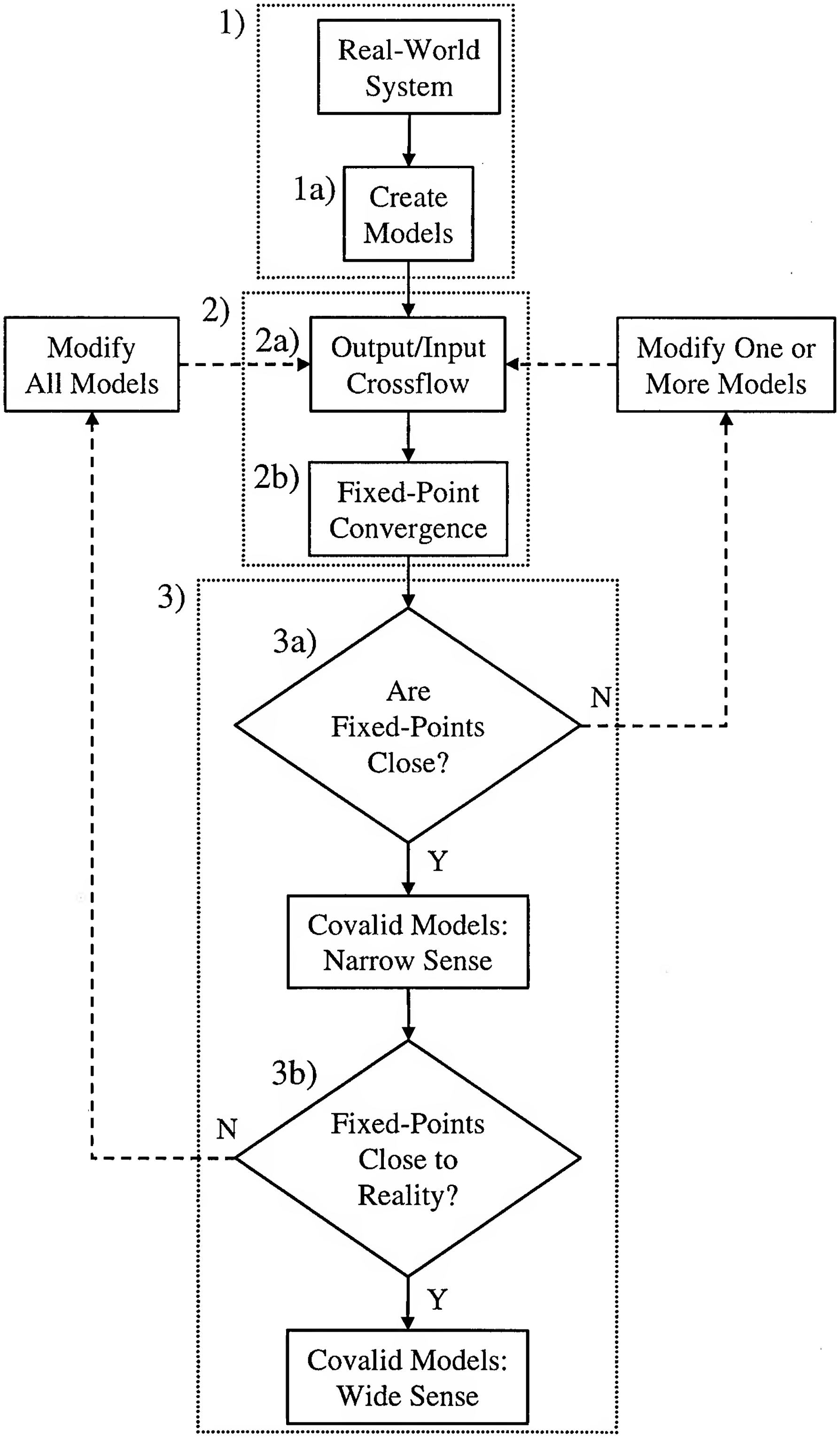

Figure 2: General Methodology

Contributions

This research provides significant contributions to the general field of mathematical modeling. An original paradigm is established for the construction, improvement, and validation of models. Within this framework, the original notion of the covalidation of models is detailed. First, an iterative method is devised that seeks the improvement of the inputs to each model considered. Further, it is proven that this method converges to fixed-point input values under certain assumptions. Second, a method of gradient analysis is devised which allows the direct comparison of the models. Also, we demonstrate the entire process on two large-scale, real-world models in use by the United States Air Force Air Mobility Command.

Background

The ultimate goal of this research is to provide a methodology by which two or more large-scale models that model the same real-world system may be compared and contrasted. The impetus for this research was interest in the comparison of two such models. One model is a large-scale discrete event simulation model that enjoys a relatively high level of acceptance. The other is a large-scale linear programming model designed to optimize the same general system modeled by the simulation.

The simulation model is used by the Air Mobility Command (AMC) of the United States Air Force and is known as the Mobility Analysis Support System (MASS). The Airlift Flow Module (AFM) is the simulation core of MASS. It simulates the movement of detailed cargo requirements through the airlift system based on the availability of aircraft, air routes, and air base infrastructure and resources. For this research, the terms

MASS and AFM are used interchangeably. The simulation is deterministic in its mission planning while mission execution is stochastic. A synopsis of MASS is given by an AMC Studies and Analysis Flight (AMCSAF) point paper prepared for the Congressional Budget Office (Merrill, 1993):

> Inputs. Inputs to the MASS include:
>
> - A Time-Phased Force Deployment Data (TPFDD) document containing airlift movement requirements
> - An airlift network of onloads, offloads, en-route stops, recovery bases, and home stations connected by user-defined routes
> - An airlift fleet mix of different aircraft types identified by individual tail numbers
> - Individual aircrews who must be available to allow missions to be flown
> - Logistics factors which account for refueling, maintenance, and material handling of cargo
> - Concepts of operations that include strategic intertheater airlift, aerial refueling, intratheater shuttle operations, direct delivery operations, and recovery/stage operations
>
> Planning. Mission planning in the AFM accomplishes:
>
> - Prioritization of requirements by available-to-load dates and required delivery dates
> - Prioritized route selection and reservation for flight planning
> - Marrying a specific aircraft tail number to the next eligible requirement
> - Crew planning to ensure that only the crews eligible to fly do fly
>
> Execution. Mission execution in the AFM simulates:
>
> - Typical sortie events, including: taxi-out, takeoff, departure, en-route cruise, initial approach, final approach, landing, taxi-in, and ground activities for every sortie of the mission

- Ground activity resource allocation and planned delays for: ramp space, offloading cargo, refueling, maintenance, onloading cargo, and crew changing

- Optionally, detailed loading of each piece of cargo for compatibility with doors and remaining space on each aircraft

- Crew activities and monitoring events, including: crew rest, crew monthly and quarterly flying hour limits, crew availability, and searches for unavailable crews

Output. Output from the AFM includes:

- Aircraft-related statistics, such as: utilization rate, payload, ground service time, flight time, and system delays

- Aircrew related statistics, such as: crew duty day, number of crews, hours flown by each crew, and crew availability

- Cargo related statistics: total tons delivered, tons per day throughput, unit and force closure, actual million tons miles per day flown, and cargo remaining in backlog

- Airlift network statistics: typical cycle times, flying times, network airfield use, maximum on the ground (MOG) constraints, and system bottlenecks

The other model is a large-scale optimization model developed jointly by the Naval Postgraduate School (NPS) and the RAND Corporation known as the NPS/RAND Mobility Optimizer (NRMO) (Morton, Rosenthal, and Weng 1996). NRMO models the strategic airlift system as a multi-period, multi-commodity network-based linear program (LP) with many side constraints. Use of the model is intended to provide insight into mobility problems such as fleet and infrastructure adequacy, and the identification of system bottlenecks. A summary of the model's characteristics follows:

Inputs. Inputs to the NRMO include:

- A TPFDD containing airlift movement requirements

- An airlift network of onloads, offloads, en-route stops, recovery bases, and home stations connected by allowable routes

- An airlift fleet mix of different aircraft types and their characteristics, delineated by the numbers of each type

- Numbers of aircrews available to perform missions at each base

Constraints. Constraints in the NRMO ensure:

- Proper aircraft allocation and balance at all nodes of embarkation and debarkation

- Proper aircrew allocation and balance of flow

- Aircraft do not fly more missions than their utilization rate allows

- Demand is met for each line of requirement

- Airfield capacity is not exceeded

Objective Function. The objective function in the NRMO minimizes:

- Amount of non-delivered cargo times a weight factor

- Amount of cargo delivered late times a weight factor

- Penalty for reassigning aircraft missions, (negative) bonus for aircraft remaining at home station (i.e., maximize the bonus), penalty for crews forced to deadhead, each multiplied by a weight factor

Output. Decision variables from the NRMO include:

- Aircraft-mission statistics: the number of aircraft delivering a particular cargo over a route at a particular time or the number of aircraft recovering from such a mission at a particular time

- Aircraft inventory statistics: the number and type of aircraft remaining over night at a base

- Aircraft changing roles: numbers of new aircraft allocations or aircraft changing roles (cargo hauling to tanker, or tanker to cargo hauling)

- Cargo related statistics: tons of each type of cargo delivered by a particular aircraft at a particular time

- Crew statistics: number of crews available for an aircraft type at a base at a particular time and number of deadheading crews

NRMO's minimization objective function is the weighted sum of three sub-objectives that seek to minimize the non-delivery of cargo, the lateness of cargo deliveries, and the penalty for performing certain undesirable actions (such as deadheading crews). First, a large relative weight is attached to the non-delivery of cargo. Next in relative importance is minimizing the lateness of cargo deliveries. Finally, a third, relatively small weight is applied to minimizing the penalty for performing undesirable actions. The weights applied to each sub-objective are subjective, but they are generally ordered in that weight for non-delivery is far greater than the weight for late delivery that, in turn, is far greater than the weight of the (negative) bonus.

The two models have many differences as well as similarities. One similarity is that both model strategic airlift and provide certain common outputs, such as the amount of cargo delivered. The first major difference between the models is that NRMO optimizes the airlift schedule while MASS schedules flights based on the availability of aircraft and a prioritized list of routes. The second major model difference is that MASS models most event durations as random variables while NRMO employs mean values. A third major difference is that while MASS provides a detailed look at many aspects of the airlift system, NRMO represents a much more aggregate view of the airlift system.

The MASS simulation is currently in use by the AMC Studies and Analysis Flight (AMCSAF), and though a formal validation has not been accomplished, its results are generally accepted as valid. One obstacle to a traditional, output-based validation of either model is that there is no way to collect real-world data from a strategic airlift system due to the infrequency of actual large-scale conflicts. Desire by AMCSAF to have some basis for the use of the NRMO optimization model has provided a motive for comparing and contrasting the two models.

Test Models

As a lucid demonstration of the proposed methodology, very small-scale test models based on the MASS and NRMO models are developed. A small-scale simulation is constructed which models the movement of simple blocks of cargo with a fleet of identical aircraft from a single onload point to one of two offload points, after which aircraft recover to the onload point for further missions. The proportion of missions flown to each offload point is user defined.

A small linear program is also created which optimizes the amount of cargo that can be moved across the same airlift system the simulation employs. Input into this model is an estimate of the efficiency of ramp space usage, which accounts for the fact that ramp space may not be optimally scheduled in practice. The LP outputs the total amount of cargo moved as well as the amount of cargo moved to each offload point, thereby implying an optimal proportion of use for the two offload points.

Data used in these models are contrived and are not intended to resemble any actual airlift system data. Likewise, results obtained from these models are not intended to mimic those of the MASS model, the NRMO model, or those of any actual airlift scenario.

Organization

The remainder of this dissertation is organized as follows. Chapter II is a literature review that provides a background to several techniques used in this research. Concepts such as fixed-point analysis, metamodeling, and model comparison are discussed here, and a brief coverage of validation literature is offered. Chapter III details the methodology developed to conduct this research, including an iterative technique designed to exploit the crossflow of outputs and inputs between dissimilarly structured models as well as a gradient analysis approach to model comparison. In Chapter III, an overview of the methodology is given and I describe how to apply it. Then, the theoretical basis for output/input crossflow is provided. This is followed by a small example that demonstrates the use of the methodology on small-scale test models. In Chapter IV, the method's performance on the MASS and NRMO models is detailed. First, a description of the scenarios used in this research is given. Then the actual performance of the models with regard to the developed method is described. Finally, I conclude the research in Chapter V with a summary and recommendations for further research.

II. Literature Review

Introduction

The body of literature concerning the comparison of dissimilarly structured models is small. However, the methods developed here for making such comparisons rely largely on the adaptation of other well-known techniques. These techniques include fixed-point analysis, experimental design, metamodeling, sensitivity analysis, model validation, and model comparison. Literature concerning fixed-point analysis are reviewed in the next section. Relevant literature pertaining to experimental design, metamodeling, and sensitivity analysis are presented in the following section. Selected literature concerning validation techniques is reviewed in the next to last section. Model comparison literature is covered in the last section.

Fixed-Point Analysis

The first major step in implementing the methodology developed here involves the convergence of input and output vectors to a fixed point at which I assert two or more models are behaving as nearly alike as possible. Works found to be of use in developing the proof that not only do fixed points exist given certain assumptions, but that under other conditions they may be unique and relatively simple to locate include those by Apostol (1974), Border (1985), Istratescu (1981), and Smart (1974).

Border, Istratescu, and Smart all provide coverage of Brouwer's Fixed Point Theorem which establishes that at least one fixed point exists for a mapping which maps a convex and compact subset of *m*-space to itself (Border 1985; Istratescu 1981; Smart 1974).

Further, Apostol, Istratescu, and Smart give accounts of how to find a unique fixed point in the set if the mapping is a contraction mapping (Apostol 1974; Istratescu 1981; Smart 1974). While this work makes the assumption of a contraction mapping, Border, Istratescu, and Smart all provide solutions to determining fixed points given the mapping is not found to contract.

Metamodeling

Kleijnen and Van Groenendaal (1992) describe the desire for the creation of a simpler representation of the real world than even a simulation model presents, a sacrifice of accuracy for expedience. Simulations, themselves merely models of an actual system, only give responses at a limited number of selected input combinations. The number of responses possible is determined by the amount of computer time available. Creation of a regression function (metamodel) for some inputs of interest (for some interesting output variable) allows the interpolation and (to a lesser degree) extrapolation of the inputs in order to predict an output. These inputs of interest could consist of input variables, which can be observed directly in the real system (as the amount of a resource), or input parameters which cannot be directly observed (as a service rate or an efficiency factor). They demonstrate the creation, estimation, and validation of metamodels for both stochastic simulations and deterministic simulations. Though they do not explicitly describe these techniques for optimization models, optimizations may be thought of as deterministic simulations, since the requirement is that for some input to the model, the output of the model is deterministic, with zero variance (Kleijnen and Van Groenendaal 1992).

In the estimation of metamodel parameters, Kleijnen and Van Groenendaal (1992) suggest the use of ordinary least squares, estimated weighted least squares, or corrected least squares regression techniques, depending on the form of the covariance matrix of the output variable. Should the experimental input vectors be relatively close (i.e., a small design space), ordinary least squares regression should suffice. For the validation of metamodels, Kleijnen and Van Groenendaal suggest the simultaneous (Bonferroni) testing that the new observations' Studentized prediction error is within the 1 - $\alpha/2$ quantile of the standard normal variable (Kleijnen and Van Groenendaal 1992).

Van Groenendaal and Kleijnen (1996) further discuss the use of an effective design of experiments in order to minimize the number of required simulation runs. Their point here is that design of experiments combined with regression metamodels provides a more sound method of sensitivity analysis than changing one factor at a time or simulating a few random samples, while still being simple to implement (Van Groenendaal and Kleijnen 1996).

The choice of an experimental design is a rather subjective issue which is determined by, among other things, the number of runs possible (access to computer time), the number of input factors, the desired resolution of the design, and which factor interactions are interesting. Full factorial designs vary each factor to two different levels about the design center, and the center point may be added, as well. Fractions of the full factorial designs may be used to reduce the number of runs required. However, the resolution of the design also decreases, thereby decreasing the number of estimated interactions possible. Kleijnen advocates the use of resolution III or resolution IV designs when performing simulation sensitivity analysis, depending on the requirements of the study. No matter which specific design is used, however, Kleijnen suggests factorial designs since their orthogonality yields unbiased estimators with small variances (Kleijnen 1996). Other techniques for reducing the number of required runs are Plackett and Burman designs and the use of Taguchi's methods. These methods, as well as complete overviews of experimental design are covered in detail in Box and Draper (1987) and Montgomery (1991).

Taylor, Auclair and Mykytka (1995) propose that the quality of an estimated metamodel is affected much more by the metamodel specification than by the number of replications performed to make the parameter estimates. In their study of M/M/k queues, it was determined that the proper specification of the metamodel (e.g., linear or multiplicative) had a dramatic impact on the validity of the metamodel, while efforts consisting of lengthening simulation runs or adding more simulation replications were generally not worthwhile (Taylor, Auclair, and Mykytka 1995).

Johnson, Bauer, Moore, and Grant (1996) describe the use of response surface methodology (RSM) and kriging techniques to estimate the value of the optimal objective function of a linear program over a range of right-hand-side values that may encompass multiple critical regions. The result is a simple method for performing optimality analysis while at the same time gaining insight into the relationships between the objective function value and the right-hand-side vector over the specified range (Johnson *et al.* 1996). The significance of their study is that creating metamodels of optimizations can be an effective technique, even over multiple critical regions where it is known that the overall function is not linear, but piece-wise linear.

The validity of using metamodels for simulation models is addressed by Friedman and Pressman (1988). They create metamodels for three different simulation models and conclude that the simulation metamodel is indeed a valid analysis technique for simulation models. While they show that the metamodels are relatively stable across different streams of random number inputs, they also suggest creating at least two metamodels based on separate output data in order to verify the consistency of a given simulation model (Friedman and Pressman 1988).

Validation

In order for models to be useful, they should reflect the real-world system to the degree to which they are designed. Model validation is defined as "substantiating that the model, within its domain of applicability, behaves with satisfactory accuracy consistent with the study objectives" (Balci 1994). There are a myriad of techniques available to assess the validity of simulation models, and many of these techniques may be applied to more general classes of models, such as mechanistic or optimization models. These techniques include both objective and subjective tests of models, assessing the validity of model assumptions, model structure, behavior of model execution, and the model's output performance, among other tests. Depending on the study, different aspects of model validation may be of paramount importance. Though real-world strategic airlift system data is not readily available or attainable, this effort concentrates on models' output performances.

Balci (1994) suggests validation techniques based on the input and output of simulations. First, black-box testing is used to assess the accuracy of the input-output transformation. However, he states, just because 1,000 input values are tested does not imply high confidence in the model's accuracy since 1,000 input values probably accounts for only a very small fraction of the possible input values. The higher the percentage of input values tested, the greater the confidence placed on the assessed accuracy of the input-output transformation. Separately, Balci proposes that sensitivity analysis be performed in order to ensure the most influential input parameters and variables are set as accurately as possible. Balci also provides a validation procedure based on simultaneous confidence intervals in order to account for simulation models in which multivariate responses are of interest (Balci 1994).

Sargent (1996b) provides an overview of the simulation validation process. He stresses that if a simulation is constructed to answer a variety of questions, the validity of the simulation needs to be addressed with respect to each of the questions, generally requiring several sets of experimental conditions to cover the domain of the model's intended use. For its intended purpose and over its intended domain, operational validity may be obtained if the simulation's output behavior accurately models the actual system. Sargent suggests comparing the output of the simulation model to those determined analytically (if possible) or to another (validated) simulation. He also reiterates Balci's call for sensitivity analysis in order to determine which parameters should be most accurately modeled and suggests comparison to the sensitivities of the real-world parameters. Sargent cautions that operational validity cannot be claimed, however, unless the simulation is tested with at least two different sets of experimental conditions (Sargent 1996b).

Kleijnen (1995b) provides another overview of the validation process. Of additional interest here, Kleijnen asserts that sensitivity analysis is very important when establishing the credibility of simulations which model real-world systems with unobservable outputs. This sensitivity analysis should be used to validate the reaction of the output to changes in the input, as predicted by system experts (Kleijnen 1995b). In a related article, Kleijnen (1995a) also stresses that it is difficult to mimic all the relevant factors of a real-world system in a simulation model and that the resulting differences may yield simulation inaccuracies in addition to those related to the stochastic nature of the simulation (Kleijnen 1995a). The relevance of this assertion to the situation of comparing two differently structured models is clear, and though the input parameters in models may be completely specified, care must be taken in order to assure they are consistent between the models.

While nearly all the literature concerning model validation focuses directly upon validation of simulation models, Adelman (1992) reviews experimental validation techniques which have been applied to structural optimization problems. He laments the lack of real-world optimization validation examples and suggests the lack of such examples may be one reason for the apparent lack of interest in optimization validation. Adelman suggests that validating an optimization consists of (1) analysis correlation: establishing the accuracy of the underlying analysis (optimization objective function and constraints) and (2) design validation: comparing the results of the optimization procedure with conventional (existing) results or with test (experimental) case results. Three questions which are answerable using experimental studies are (1) did the optimization produce a solution with improved performance compared to some baseline,

(2) did the tests verify the predicted (optimized) performance of the design, and (3) are any designs in the neighborhood of the predicted optimum better than the predicted optimum (Adelman 1992).

Model Comparison

The literature is scarce concerning the comparison of two (or more) response surfaces. There is more literature available concerning the comparison of mechanistic models (Box and Hill 1967; Box, Hunter, and Hunter 1978; Hunter and Reiner 1965) or the comparison of simulation models (Law and Kelton 1991). These topics are not without parallel relative to the present topic. Further, the use of similarly structured models (specifically, simulations) has been proposed to assist in establishing model credibility (Diener, Hicks, and Long 1992).

Hunter and Reiner (1965), improved upon by Box and Hill (1967), suggest a method of model improvement in which successive input design points are added to a mechanistic model based on maximizing the difference in response of the two (current and updated) models. While this study is concerned with comparison of existing models, Hunter and Reiner's method of maximum model displacement could be applied to two (or more) existing models being compared to a reference (Hunter and Reiner 1965).

Box and Hunter (1962) and Box, Hunter, and Hunter (1978) provide ideas for testing models based on the premise that "in an adequate model constants stay constant when variables are varied" (Box and Hunter 1962; Box, Hunter, and Hunter 1978). The method suggests the recalibration of parameters across an independent variable. If the recalibration yields the same values for the parameters across the independent variable's domain, the model is determined to be adequate. This may not be practical for a response surface of many independent variables, unless insightful information concerning these variables is known in advance of the study. Their point is, however, that in order to adequately test a model, it must be put in jeopardy across important ranges of variables. This implies using data other than the data that was used to create the model for validation testing.

Law and Kelton (1991) suggest comparisons based on constructing confidence intervals concerning the difference in responses between two competing simulation models. The confidence intervals are created using either a paired-t or a modified two-sample-t method. Both methods require independent, identically distributed (IID) observations from each model, but for the paired-t test, replications between the two models need not be independent. For testing that the models are the same, if the confidence interval constructed about the response (at an appropriate level of significance) contains zero, the difference between the two models (for that response) cannot be said to be significant.

When analytic methods or real-world data are not available, Diener, Hicks, and Long (1992) suggest the qualitative and quantitative comparison of similar (simulation) models. They present a methodology, specific to simulations which model the same situation, which qualitatively compares the models based on 1) simulation background and documentation, 2) simulation features and input database, and 3) simulation model usability. For the quantitative comparison, first a measure of merit is chosen which both models can provide and which can supply a meaningful measure, consistent with their purpose. Next, common input structures must be created for each model. Then, common experimental factors are chosen. An experimental design is chosen and experimental trials are run. Lastly, statistical tests are run on the gathered data. Specifically, a paired difference test is run to determine whether a difference in output or a difference between identical treatments exists between the simulations. The application of their technique is not intended to validate a simulation as much as to increase its acceptability (Diener, Hicks, and Long 1992).

The preceding literature forms a basis for the methodologies presented in the next chapter. The process of metamodeling using response surfaces and regression techniques allows the formation of a common model type in order to apply model comparison methods. Known validation techniques along with other views concerning model comparison provides a sound basis for the comparison of differently structured models.

III. Methodology

Overview

This chapter details the methodology developed to conduct this research, including an iterative technique designed to exploit the crossflow of outputs and inputs between dissimilarly structured models, as well as a gradient analysis approach to model comparison. First, key concepts used in the development of the methodology are defined. Then, the crux of the chapter follows the blueprint laid out in Figure 1. A framework for the development of multiple models from a real-world system is offered. This section represents the formalization of block 1 from Figure 1, and it provides a foundation for the crossflow and gradient analysis methods. Then, the theoretical basis for the method of output/input crossflow is provided. This section is represented by block 2 in Figure 1 and provides a mathematical development of the method as well as a proof that the crossflow can result in the convergence to a single set of inputs for the models in question. The next section describes the use of gradient analysis to determine the extent of models' covalidity. This section is represented by block 3 in Figure 1. This chapter concludes with an example that demonstrates the use of the methodology on small-scale test models.

Modeling Vocabulary

A Mapping Matrix for Multiple Models. The general theory of mappings defined on models needs the concept of groups. Let G be a group of objects with $\oplus$ as a binary operation defined on G. The definition gives the required properties.

Definition. Let G be a nonempty set with equality (=) defined. Let $\oplus$ be a binary operation such that:

1. G is closed with respect to $\oplus$, that is, for all $g_1, g_2 \in G$ then $g_1 \oplus g_2 \in G$;
2. G is associative with respect to $\oplus$, that is, for all $g_1, g_2, g_3 \in G$ then $(g_1 \oplus g_2) \oplus g_3 = g_1 \oplus (g_2 \oplus g_3)$;
3. G has a unique identity with respect to $\oplus$, that is, there exists $e \in G$ such that for all $g \in G$ then $g \oplus e = g$ and $e \oplus g = g$; and
4. G has a unique inverse with respect to $\oplus$, that is, given $g_1 \in G$ there exists a unique $g_2 \in G$ such that $g_1 \oplus g_2 = e$.

I will be using mappings on groups. Let G and H be two groups, and let $A: G \rightarrow H$ be a mapping with domain of A equal of G, denoted by $\mathfrak{D}(A) = G$.

Let $G_1, G_2, \ldots, G_m$ and $H_1, H_2, \ldots, H_n$ be groups. For each pair of indices $(i, j) \in \{1,2,\ldots,n\} \times \{1,2,\ldots,m\}$, define the mapping $A_{ij}: G_j \rightarrow H_i$. The output of the mapping may be written as $A_{ij}(g_j)$. For each m-tuple of objects $\mathbf{g} = (g_1, g_2, \ldots, g_m) \in G_1 \times G_2 \times \cdots \times G_m$, the mapping $\mathbf{A}_i : G_1 \times G_2 \times \ldots \times G_m \rightarrow H_i$ is defined as

$$\mathbf{A}_i(\mathbf{g}) = A_{i1}(g_1) \overset{i}{\oplus} A_{i2}(g_2) \overset{i}{\oplus} \cdots \overset{i}{\oplus} A_{im}(g_m) \tag{1}$$

In Equation (1), $\overset{i}{\oplus}$ denotes the particular binary operation for H_i.

I define the n by m matrix of mappings $\mathbf{A}: G_1 \times G_2 \times \cdots \times G_m \rightarrow H_1 \times H_2 \times \cdots \times H_n$ acting on a vector of inputs $\mathbf{g}$ as

$$\mathbf{A}(\mathbf{g}) = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} = \begin{bmatrix} A_{11}(g_1) \overset{1}{\oplus} A_{12}(g_2) \overset{1}{\oplus} \cdots \overset{1}{\oplus} A_{1m}(g_m) \\ A_{21}(g_1) \overset{2}{\oplus} A_{22}(g_2) \overset{2}{\oplus} \cdots \overset{2}{\oplus} A_{2m}(g_m) \\ \vdots \\ A_{n1}(g_1) \overset{n}{\oplus} A_{n2}(g_2) \overset{n}{\oplus} \cdots \overset{n}{\oplus} A_{nm}(g_m) \end{bmatrix} \tag{2}$$

Obviously, this is not matrix multiplication but the action of the mappings on the objects in the group. The entries of the objects may be scalars or vectors in some vector space. The entries of the n by m matrix of mappings each function such that entry (i, j) in the mapping matrix is a function that maps the object of the j^{th} group, G_j, to the element of the i^{th} group, H_i. This places two restrictions on each element of a mapping matrix. First, every entry in the j^{th} column must accept as input the object found as the j^{th} element of the m-dimensional vector $\mathbf{g} = (g_1, g_2, \ldots, g_m)^{\mathrm{T}}$. Second, every entry in the i^{th} row must provide output compatible with the i^{th} element of the n-dimensional vector $\mathbf{h} = (h_1, h_2, \ldots, h_n)^{\mathrm{T}}$. Executing the mapping matrix on an input vector will occur similar to the function of normal matrix-vector multiplication except instead of multiplying the (i, j)

entry in the matrix by the j^{th} entry in the vector, the mapping at the (i, j) entry in the matrix acts on the j^{th} entry in the vector.

The special mapping that maps all objects in G_j to the identity object in H_i will be denoted by 0, the "zero" mapping. If $H_i = G_j$, then the special mapping that maps g to g is called the identity mapping and is denoted by I.

An example of how a mapping matrix operates is given in Equation (3).

$$\begin{aligned}
\mathbf{A}(\mathbf{g}) &= \begin{bmatrix} A_{11} & I & A_{13} \\ 0 & A_{22} & I \\ A_{31} & 0 & A_{33} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \\
&= \begin{bmatrix} A_{11}(g_1) \overset{1}{\oplus} g_2 \overset{1}{\oplus} A_{13}(g_3) \\ 0 \overset{2}{\oplus} A_{22}(g_2) \overset{2}{\oplus} g_3 \\ A_{31}(g_1) \overset{3}{\oplus} 0 \overset{3}{\oplus} A_{33}(g_3) \end{bmatrix} \\
&= \begin{bmatrix} A_{11}(g_1) \overset{1}{\oplus} g_2 \overset{1}{\oplus} A_{13}(g_3) \\ A_{22}(g_2) \overset{2}{\oplus} g_3 \\ A_{31}(g_1) \overset{3}{\oplus} A_{33}(g_3) \end{bmatrix}
\end{aligned} \tag{3}$$

Note that if there are more than one non-zero entries in a row of the mapping matrix, the format of those entries' output must be compatible with the binary operation denoted by the particular $\oplus$ symbol. (Throughout this research, I use vector addition for this operation.) Further, if one of those entries is an "*I*", the format of the output must be the same as that of the input to accommodate the element-by-element summation. In Equation (3), for instance, $H_1 = G_2$ and $H_2 = G_3$.

Aggregation of Models. I define the *aggregation* as the reduction of a countable number, *n*, bits of information (descriptors of reality) to *m* bits of information, where $m < n$. Since one cannot perceive reality at the finest granularity possible, I assume that all transformations of information from reality to the mathematical realm include some level of implicit aggregation. I consider this implicit rather than explicit aggregation since it is considered implausible to create a mathematical representation at the finest granularity of reality, if such a representation exists, and aggregate from there.

A single state of reality may be aggregated in more than one manner, each manner valid for its specific purpose. For example, suppose reality includes, in part, the fact that in parking space "1" at a given base "A" there stands a C-141, gray in color, possessing a specific tail number, interior cargo volume equal to its interior length times its width times its height, with one slightly loose bolt on one of its wheels, and clearly having a multitude of other possible levels of description. Also, in parking space "2" at the same base, is a C-5 with a set of attributes comparable to those for the C-141. A person interested in modeling a system which includes these aircraft (assumed to be the only aircraft at base A) may express the information as "number of C-141s at base A = 1 and number of C-5s at base A = 1." The expression of the properties of aircraft into mathematical values is an example of implicit aggregation. Alternatively, a more generic model may require translation of the information concerning the aircraft at base A as "number of aircraft at base A = 2." Other models may require much more detail regarding the reality of aircraft at the base.

Assume two mathematical representations exist for the same system and that all information in representation "2" may be derived through aggregations of the information found in representation "1". The term *more aggregate* will be used to describe the aggregation level of representation "2" relative to representation "1". The term *less aggregate* (or equivalently, *more disaggregate*) will be used to describe the aggregation level of representation "1" relative to representation "2". Note that it is possible in a mathematical representation for certain information to exist at a more aggregate level while other information exists at a less aggregate level than in another mathematical representation. In such a case, the terms more aggregate, less aggregate, or more disaggregate will not be used to describe the relative aggregation levels of the representations. Rather, the representations may be said to have *mixed aggregations* (with respect to each other). The terms more aggregate, less aggregate, and more disaggregate, however, may be used in describing the specific information (within the representations) for which they apply.

Modeling the Real World

"A *system* is a collection of items from a circumscribed sector of reality that is the object of study or interest" (Pritsker 1986). "A system is defined to be a collection of entities, e.g., people or machines, that act and interact together toward the accomplishment of some logical end" (Law and Kelton 1991). Pritsker's definition chisels a system from a larger picture, namely reality, while Law and Kelton build a system from smaller components. I acknowledge that either description adequately defines a system as referred to here. A system may be as simple as a single-server barbershop or as complex as the global economy, of which the single-server barbershop may be a part. A system may have a logical beginning and end, such as the opening and closing times of the barbershop, or may be on-going without a definite beginning or an obvious end, as in the global economy.

A mathematical model is a mathematical description of a system, and is referred to here simply as a model. Models represent the system "in terms of logical and quantitative relationships" that are "manipulated and changed" to determine the model's reaction, and therefore the system's reaction—if the mathematical model is valid (Law and Kelton 1991). (Since the systems here are of a temporal nature, I relate the representations of these systems to models of a temporal nature. Though a system's (and therefore its model's) temporality is not necessary for the devised methodology to succeed, the paradigm is easily illustrated under this assumption.) A model can represent a system from its beginning to its end, or only part of the system could be modeled (i.e., half of a day at the barbershop, or a year of the global economy). Either way, a model is assumed to have a definite beginning and end, at which points the state of the model may be sampled (regardless of the ability to sample the system at these points), revealing the model's input and output, respectively.

The Real World and Real-World Systems. Define $\mathcal{X}_{\text{real}}$ to be the condition of reality at some system onset time ($t = 0$). $\mathcal{X}_{\text{real}}$ includes the exact position of every physical thing, the true capabilities and limitations of every physical thing, the physical, mental and emotional state of every individual, all plans, as well as the state of any other thing at $t = 0$. In addition to this, all information available at $t = 0$ is part of $\mathcal{X}_{\text{real}}$ (regardless of whether such information is actually known or not). This includes the entire history of reality (i.e., the state of reality at all times $t < 0$) and all physical, mental, emotional, and other actual laws (known or unknown) which govern the universe.

Define $\mathcal{Y}_{\text{real}}$ to be the collective states of reality at every time $t > 0$ until some system termination time (t = termination). $\mathcal{Y}_{\text{real}}$ includes the exact position of every physical thing, the true capabilities and limitations of every physical thing, the physical, mental and emotional state of every individual, all plans and intentions, as well as the state of any other thing at every time $0 < t \leq$ termination. In other words, whereas $\mathcal{X}_{\text{real}}$ is the state of every component of reality at $t \leq 0$, $\mathcal{Y}_{\text{real}}$ is that state at every $0 < t \leq$ termination.

$\mathcal{T}$ is defined as everything that happens between time "zero" and time "termination" which causes changes in the state of reality. The execution of all the laws (known or unknown) which govern reality and the choices individuals make is considered part of $\mathcal{T}$.

$\mathcal{X}_{\text{real sys}}$ is defined as the state of a system at all times up to and including system onset time ($t \leq 0$). $\mathcal{X}_{\text{real sys}}$ is merely the subset of information from $\mathcal{X}_{\text{real}}$ that defines that circumscribed sector of reality that is the object of study. It is clear that $\mathcal{X}_{\text{real sys}} \subset \mathcal{X}_{\text{real}}$, and I define the function which selects the subset $\mathcal{X}_{\text{real sys}}$ as $\mathcal{S}_{\mathcal{X}}$. After each general definition for a system or model component, an example illustrative of my specific application, namely the strategic airlift system, is given. Consider the strategic airlift system. The initial set of information that describes this system is $\mathcal{X}_{\text{real sys}}$ and includes such things as the exact position of every physical thing which influences the strategic airlift system, including cargo, infrastructure, equipment, resources, personnel, atmospheric conditions, the true capabilities and limitations of every piece of equipment in the strategic airlift system, the physical, mental and emotional state of each individual having anything to do with the strategic airlift system, the plans for the movement of cargo, the flights of aircraft, and the scheduling of crews, as well as the state of any other thing which could be involved in the strategic airlift system at $t = 0$. In addition, all information concerning the strategic airlift system available (whether known or not) at $t = 0$ is part of $\mathcal{X}_{\text{real sys}}$. This includes the history of the strategic airlift system and all physical, mental, emotional, and other laws (known or unknown) which govern the strategic airlift system.

$\mathcal{Y}_{\text{real sys}}$ is defined as the collection of system states at every time $t > 0$ until some system termination time. Again, $\mathcal{Y}_{\text{real sys}} \subset \mathcal{Y}_{\text{real}}$ and the function which selects the subset $\mathcal{Y}_{\text{real sys}}$ is defined as $\mathcal{S}_{\mathcal{Y}}$. $\mathcal{Y}_{\text{real sys}}$ for the strategic airlift system includes such things as the exact position of every physical thing which influences the strategic airlift system, including cargo, infrastructure, equipment, resources, personnel, and atmospheric conditions, the true capabilities and limitations of every piece of equipment in the strategic airlift system, the physical, mental and emotional state of each individual having anything to do with the strategic airlift system, the movement of cargo, the flights of aircraft, and the actual scheduling of crews, as well as the state of any other thing involved in the strategic airlift system at every time $0 < t \leq$ termination. Whereas $\mathcal{X}_{\text{real sys}}$ is the state of every component of the strategic airlift system at $t \leq 0$, $\mathcal{Y}_{\text{real sys}}$ is that state at every $0 < t \leq$ termination.

$\mathcal{T}_{\text{sys}}$ is defined as everything that affects the system (i.e., modifies or acts on $\mathcal{X}_{\text{real sys}}$) between the defined onset and termination times. The execution of all the laws which govern the system and the choices made by individuals involved in the system are considered part of $\mathcal{T}_{\text{sys}}$. The execution of all physical laws concerning the strategic airlift system—the loading of cargo, the flying of aircraft, the changing of the weather, and the failure of equipment—are considered part of $\mathcal{T}_{\text{sys}}$. Also, decisions of individuals involved in the strategic airlift system—the decisions to execute the airlift plans, to fly an aircraft, whether to climb above the cloud or fly through it, whether a pilot gets out of bed or not when her alarm goes off in the morning—are all considered part of $\mathcal{T}_{\text{sys}}$. Figure 3 presents a graphical description of these relationships.

$$\begin{tikzcd}
\boxed{\mathcal{X}_{\text{real}}} \arrow[r, "\mathcal{T}"] \arrow[d, "\mathcal{S}_{\mathcal{X}}"'] & \boxed{\mathcal{Y}_{\text{real}}} \arrow[d, "\mathcal{S}_{\mathcal{Y}}"] \\
\boxed{\mathcal{X}_{\text{real sys}}} \arrow[r, "\mathcal{T}_{\text{sys}}"] & \boxed{\mathcal{Y}_{\text{real sys}}}
\end{tikzcd}$$

Figure 3: Real World and Real-World System Relationship

Representing a Real-World System Mathematically. In order to create a model for a real-world system, the system needs to be represented with more mathematically meaningful terminology than is found in the definitions of $\mathcal{X}_{\text{real sys}}$, $\mathcal{T}_{\text{sys}}$, or $\mathcal{Y}_{\text{real sys}}$. What is required is a method of characterizing objects that exist in reality with mathematical representations. Of course, these representations are generally created through the implicit aggregation performed mentally (even subconsciously) by those interested in a particular system.

I define X_T as an countable-dimensional mathematical representation of $\mathcal{X}_{\text{real sys}}$. Also, $\mathcal{P}_{\mathcal{X}}$ is defined as the mapping from $\mathcal{X}_{\text{real sys}}$ to an X_T. Clearly, the number of possible X_T's which represent a particular system is very large, and each can be thought to have been mapped separately from $\mathcal{X}_{\text{real sys}}$ (through its own "$\mathcal{P}_{\mathcal{X}}$"). I define the collection of all functions that translate $\mathcal{X}_{\text{real sys}}$ to specific X_T's as $\mathcal{P}_{\mathcal{X}}$. The collection of all possible mathematical representations X_T that may be translated from a specific $\mathcal{X}_{\text{real sys}}$ is defined as $\mathbf{X}_{\text{rep}}$. A scenario is defined as a specific set of input values required for the execution of a model. An element of X_T representing a scenario is labeled x_T.

Y_T is similarly defined as a countable-dimensional mathematical representation of $\mathcal{Y}_{\text{real sys}}$. I use y_T to represent the output of a model from a specific scenario ($y_T \in Y_T$). Here, $\mathcal{P}_{\mathcal{Y}}$ is defined as the mapping from $\mathcal{Y}_{\text{real sys}}$ to Y_T. Again, the number of possible Y_T's which represent a system is very large, and each can be thought to have been mapped separately from $\mathcal{Y}_{\text{real sys}}$ (through its own "$\mathcal{P}_{\mathcal{Y}}$"). I define the collection of all functions that translate $\mathcal{Y}_{\text{real sys}}$ to specific Y_T's as $\mathcal{P}_{\mathcal{Y}}$. The collection of all mathematical representations Y_T translated from a specific $\mathcal{Y}_{\text{real sys}}$ is defined as $\mathbf{Y}_{\text{rep}}$. $\mathbf{Y}_{\text{rep}}$, then, is the collection of every aggregation of the information found in $\mathcal{Y}_{\text{real sys}}$. For the strategic airlift example, this could include not only that a particular piece of cargo has traveled from wherever it was at $t = 0$ along some path during $0 < t <$ termination to wherever it is at $t =$ termination (if some model also provides information at that level of disaggregation), but also that it has moved a certain number of miles, that a certain amount of bulk-sized cargo moved so many miles, that a certain amount of cargo has moved successfully, that a certain requirement had a particular closure time (earliest time at which all cargo in the requirement has been moved), as well as that a certain percentage of the delivery requirements were met.

T is defined as a "truth model" of the system. T maps a specific X_T into a specific Y_T. Its exact form is probably unknown. However, if a system is observable in some manner, a mathematically meaningful representation of the system input (an X_T) which is mapped (via T) into a mathematically meaningful representation of the system output (a Y_T) can be seen. Obviously, any X_T may be mapped into any Y_T, regardless of causality. Therefore, the collection of all possible truth mappings is defined as **T**. Following the strategic airlift example, though we cannot define the mathematical form of a desired T, we could say that a particular T, say T_1, maps some number of available C-5s and C-17s into some values for "bulk tons moved," "over-sized tons moved," and "out-sized tons moved." Another T, perhaps T_2, maps some number of C-5s and C-17s into the number of total tons moved. And yet another T maps total planes to the number of total tons moved. Figure 4 shows the extraction of mathematically useful information from the real world.

$$\begin{tikzcd}
\boxed{\mathscr{X}_{\text{real}}} \arrow[r, "\mathscr{T}"] \arrow[d, "\mathscr{S}_{\mathscr{X}}"'] & \boxed{\mathscr{Y}_{\text{real}}} \arrow[d, "\mathscr{S}_{\mathscr{Y}}"] \\
\boxed{\mathscr{X}_{\text{real sys}}} \arrow[r, "\mathscr{T}_{\text{sys}}"] \arrow[d, thick, "\mathscr{P}_{\mathscr{X}}"'] & \boxed{\mathscr{Y}_{\text{real sys}}} \arrow[d, thick, "\mathscr{P}_{\mathscr{Y}}"] \\
\boxed{\mathbf{X}_{\text{rep}}} \arrow[r, thick, "\mathbf{T}"] & \boxed{\mathbf{Y}_{\text{rep}}}
\end{tikzcd}$$

Figure 4: Obtaining Mathematical Representations from the Real World

Modeling a Real-World System. Given m models ($m \geq 1$) created for a system, let X_i be the set of information which is required as input to the i^{th} model, $1 \leq i \leq m$. This required input set for a desired model is available as one of the realizations of X_T. In other words, $X_i \in \mathbf{X}_{\text{rep}}$. However, all of the information that is required may not be readily mapped from reality (arrival and service rates, for example), and approximations must be made for such information. Therefore, if all information required for a model is obtainable from reality, we can say $X_i = X_T$, but often, X_i is an approximation of X_T. The input to a model representing a particular scenario is shown as x_i, where $x_i \in X_i$.

Y_i is the set of information that is derived from model i, i.e., model i's output. This set of information is available as one of the possibilities of Y_T, and so $Y_i \in \mathbf{Y}_{\text{rep}}$. However, Y_i is the output of a model and as such the values which comprise Y_i are approximations of those which comprise Y_T (for a given $\mathbf{Y}_{\text{rep}}$). The realization of a specific Y_i (in other words, the output of a model having run a specific scenario) is labeled y_i, where $y_i \in Y_i$.

M_i is defined as the i^{th} model. M_i is a mapping from X_i to Y_i which is created in an attempt to emulate the truth mapping T from X_i to Y_i. For a unique input x_i, we write the unique output as $M_i(x_i)$ or, equivalently, y_i. All variables and parameters that comprise a specific model are considered inputs to the model (X_i), and are not part of the model *per se* (e.g., service rates). However, specific requirements of a modeling type that cannot be reasonably taken from X_{rep} are considered part of the model (e.g., random number seeds). Models may be created to represent any desired level of detail and any or all parts of the system being modeled. In other words, for the strategic airlift system, a model may be created which simulates to a great level of detail the various aspects of strategic airlift. Or, a model may be created which accepts as inputs a number of aircraft and the number of days during which to fly and simply outputs throughput. Alternatively, a detailed model could be created to determine the length of time required to fly from one base to another. As each of these purposes exists in the real-world strategic airlift system, each is considered a model relative to that system, though their purposes and methods clearly differ from each other.

Determining the proximity of M_i (which maps from X_i to Y_i) to the corresponding T mapping (from X_i's corresponding X_T to Y_i's corresponding Y_T) represents an attempt at model validation. Typically, however, we do not know much about the mathematical form of T. In lieu of comparing M_i to T for validation, then, we may attempt to ensure (for a scenario) $x_i = x_T$ and determine the proximity of y_i to y_T. Complete validation would require this comparison for every possible scenario, but this is typically prohibitive so a representative subset of scenarios is generally chosen and comparisons are made from this scenario subset. A problem exists in that some or all of the information in X_T or Y_T may be difficult to obtain either directly or indirectly from observation (such as input parameters), or perhaps the truth scenario has not been (or cannot be) executed. These possibilities can make validation based on output performance difficult or impractical. If only part of X_T or Y_T is observable, a partial validation may occur over that observable part. In general then, output-based validation may take place at the intersections of X_i and the observable portion of X_T ($X_i \cap X_{T\,obs}$), and of Y_i and the observable portion of Y_T ($Y_i \cap Y_{T\,obs}$).

If two (or more) models, say M_i and M_j, are constructed for the same system, it is possible to make comparisons between the models. These attempts to determine the proximity of M_i to M_j will be labeled covalidation efforts. If the form of the models is identical (such as linear regressions in which only coefficients are different), we may compare M_i to M_j directly. However, the most likely case is that the models do not share the same form. In this case, we would like to ensure $x_i = x_j$ and determine the proximity of y_i to y_j for all scenarios $x_i \in X_i$ corresponding to $x_j \in X_j$. (Again, a representative subset of all possible scenarios will suffice.) However, it is not generally the case that the exact same objects exist in X_i (or Y_i) as X_j (or Y_j). For this reason, covalidation requires ensuring the values in each model in $X_i \cap X_j$ are equivalent, and evaluating the difference (not necessarily a strict subtraction) between the models with respect to the values of their common output in $Y_i \cap Y_j$.

Output/Input Crossflow

Introduction/Ground Rules. Here, an iterative method is employed in which selected inputs as well as outputs may converge. The purpose of this iterative scheme is to effect the output/input crossflow between the models. That is, one model's output (or a function of that output) is supplied as input to the other model. For instance, NRMO provides as output the optimal selection of aircraft routes, while MASS accepts as input the frequency of route usage. On the other hand, MASS provides output that can be translated to the efficiency of parking space use, an input parameter required by NRMO.

In dealing with dissimilarly structured models, the differences between the models must be carefully examined and exploited. Typically, dissimilarly structured models not only have different input (including both variable and parameter) sets, but they could also have different levels of aggregation as well as different capabilities in terms of modeling the actual system. In order to make a reasonable assessment of the models' covalidity, however, these differences must be examined.

Structurally different models commonly employ different levels of data aggregation. For instance, the MASS simulation models the strategic airlift system to a high level of fidelity in terms of its level of detail compared to the NRMO optimization model. When comparing such models, each model's designed level of aggregation should be maintained. In other words, the fidelity of models should not be compromised for the sake of "fair comparison." For example, optimistic optimization results could prove to be the result of unwarranted aggregation, since it may be the case that the unrealistic divisibility of aircraft or units of cargo in the optimization results in more cargo movement than is actually possible. Further, by maintaining the appropriate levels of aggregation in each model, the covalidation process may also provide information on the appropriateness of such aggregation.

Whether or not to use the different modeling capabilities inherent in each model should be carefully considered on a case by case basis before model comparison is performed. In general, the "extra" capabilities of one model compared to the other model should be switched off, if possible. For example, since NRMO can effectively model aerial refueling aircraft operations while MASS cannot, the NRMO capability should be turned off during the comparison. If this is not possible, selection of the input variables or parameters should be such that the capability exercises no significant effect.

An exception to this general rule occurs when inherent differences in the modeling paradigms used allow one model to adequately model a system aspect while the other cannot. A simple example of this is that NRMO does not model the variability inherent in many airlift processes, relying instead on mean values, while MASS models the random distributions of such variables. This difference in capabilities accounts for a fundamental difference between the two models, one for which comparisons in terms of covalidation are desired.

A difficult area is ensuring a rough parity in inputs between the two models, particularly models of different types or models that operate at different aggregation levels. Even when a particular input feeds both models, if the level of aggregation for this input is different between the two models, special care must be taken to ensure equitable representation.

After all else has been accomplished in order to ensure the models are executing the same scenario, it is likely, due to the differing natures of the modeling paradigms used, that the models will not yield the same result in terms of some common output. This is expected since, for instance, one model creates a schedule based strictly on the placement of required cargo movements on the TPFDD while the other model optimizes this schedule in order to maximize the movement of the cargo. A possible solution to this problem involves taking what is learned from the execution of one model and using it to improve the execution of the other. For this effort, I propose adapting certain outputs of each model for use as inputs to the other. Given the different world-views inherent in the models, it seems reasonable to assume that lessons learned from the output of one model may be used to improve the use of the other model. It is desired to select outputs of each model that can serve as inputs to the other. In general, it is not clear that finding such outputs to crossflow is possible. However, since optimization models provide the "best solution" while simulation models can provide estimates of system parameters, it seems reasonable that information could be meaningfully exchanged between these particular model types.

A Practical Application of the Crossflow Method. Here, an iterative scheme is presented that demonstrates the crossflow of outputs to inputs between two models, Model *i* and Model *j*. The application is described in Figure 5. In the figure, the superscript *n* denotes the current iteration number. Each model has an input set: X_i and X_j are the complete input sets required for each model. X_{ji} and X_{ij} are subsets of X_i and X_j, respectively, that represent the input derived from or modified in reaction to the other model's output (i.e.: from the output of Model *j* to the input of Model *i*, or vice versa; $X_{ji} = f_{ji}(Y_{ji})$). For the first model execution, some (subjective) nominal values are placed on these inputs that, during later iterations, will be derived from the output of the other model. Likewise, Y_i (or Y_j), is the complete output set from model *i* (or model *j*) and Y_{ij} (or Y_{ji}) is a subset of Y_i (or Y_j) that contains output which is utilized by the other model (output from the simulation to be used as input to the optimization or vice versa). The output subset used by the other model is "filtered" appropriately through the f_{ij} and f_{ji} feedback functions to make it usable as input for the other model. This filtering can be realized as a direct mathematical relationship or reflected as changes in policy or by adding model constraints.

At each iteration, a check is made to determine if the stopping criterion has been met. This criterion can be that a model's input has converged. Alternatively, successive iterations may indicate that a point of diminishing returns has been reached with this process. Either way, the final iteration inputs (denoted by "*") are deemed those which are as close to each other as possible, and they are used as the experimental design center for the ensuing model comparison.

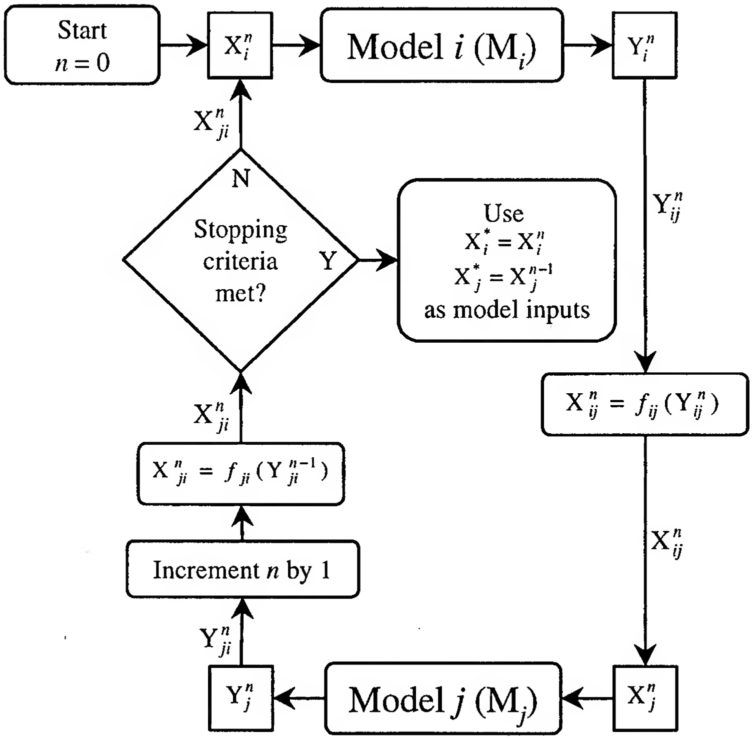

Figure 5: Iterative Scheme

In Figure 5:

n is the iteration number

<u>Model *i* input/output</u>

X_i is input to model *i*

$X_{ji} = f_{ji}(Y_{ji})$ is input to model *i* that is derived from output of model *j*

Y_i is output of model *i*

$Y_{ij} \subseteq Y_i$ is output from model *i* that is to be used as input to model *j*

Model j input/output

X_j is input to model j

$X_{ij} = f_{ij}(Y_{ij})$ is input to model j that is derived from output of model i

Y_j is output of model j

$Y_{ji} \subseteq Y_j$ is output from model j that is to be used as input to model i

Prior to developing the particulars of model comparison, I summarize the two basic steps taken to achieve a crossflow of outputs and inputs between models. First, the input/output structures of the models are carefully studied and appropriate adjustments are made due to the differences in level of detail and capability. Next, output/input links are determined which allow for a potentially meaningful crossflow.

The information provided by these links is used in an iterative fashion aimed at improving both models. This iterative scheme results in convergence to a fixed point of inputs and outputs (which may not be identical for each model). The input fixed points should represent a state where the models represent similar situations. The output fixed points can provide the first indication whether or not the models are performing analogously. The mathematical theory supporting the successful performance of output/input crossflow is begun in the following subsection.

Mathematical Development. As stated, there are likely values that are part of X_i or X_j that may only be approximated, since they are unobservable in X_T. It is possible that the output of another model may provide insight as to the true measure of such an input. The function that translates the output from model i to the input of model j is denoted by f_{ij}. Specifically, f_{ij} maps the subset Y_{ij} of Y_i to the subset X_{ij} of X_j. In general, the function f_{ij} can be thought of as mapping the entire $y_i \in Y_i$ vector to the entire $x_j \in X_j$ vector, which represents some (likely incremental) change to the original X_j. As noted, it is likely that most of y_i will not be used by f_{ij} and, correspondingly, that most of the new x_j vector will consist of values from the original x_j vector. Figure 6 describes these relationships between models and their feedbacks graphically. A dotted line in the figure denotes that a set is drawn from a collection of sets.

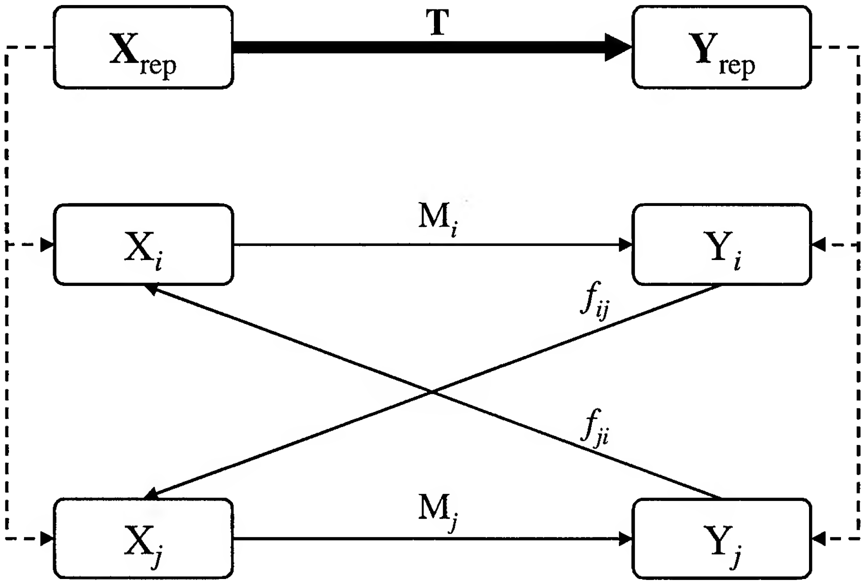

Figure 6: Relationships between Models

We wish for the opportunity of feedback between models to be more general, however. For instance, we desire to allow feedback from several models to a single model. Also, we do not wish to rule out feedback from a model to itself. For this reason, I redefine the output of the f_{ij} function as a vector that contains only the change to X_j suggested by model *i*. In other words, the majority of the $f_{ij}(Y_i)$ vector will contain "0" values, meaning that either the function does not affect these input parameters or variables or that the function does influence these parameters or variables, but no change is suggested at this iteration. The non-zero values indicate how much a particular parameter or variable is to be changed from the last iteration. In this way, more than one model (including itself) may affect another model's input. I use the notation $f_{ij}(\mathrm{y}_i)$ to denote this mapping of the output of model i to the incrementally changed x_j vector. However, $f_{ij}(\mathrm{y}_i)$ is not the only input to x_j. The entire new x_j vector is the vector sum (denoted by the $\oplus$ symbol) of each of the $f_{kj}(\mathrm{y}_k)$ vectors where $k = 1,\ldots, m$ for m models that represent the system, as shown here.

$$\mathrm{x}_j = f_{1j}(\mathrm{y}_j) \oplus f_{2j}(\mathrm{y}_j) \oplus \ldots \oplus f_{mj}(\mathrm{y}_j) \qquad (4)$$

Of course, if one of the m models (say model i) does not provide feedback to model j, then $f_{ij}(\mathrm{y}_j)$ should yield an appropriately-sized "0" vector.

It is important to note that with this definition of mapping to X_j, care must be taken that each input variable or parameter is only changed via a single model. If it is desired that more than one model affect a single input variable or parameter, another method of determining X_j (such as averaging the non-zero values for a particular element in X_j) may be developed. This work employs only vector summation in determining X_j (and the assumption that each parameter or variable is affected by at most one model).

Note that so far, the new x_j vector consists only of the change to the old x_j vector. I define the result of the feedback $f_{jj}(\mathrm{y}_j)$ as the sum of the old x_j and any desired feedback from model j to itself. In this way, the vector sum of all the $f_{kj}(\mathrm{y}_k)$ vectors (as described in Equation (4) above) now represents the exact vector we wish to use as input into model j during the next iteration. This process is illustrated in Figure 7 where the new input to model j consists of the vector summation of the $f_{kj}(y_k)$ vectors.

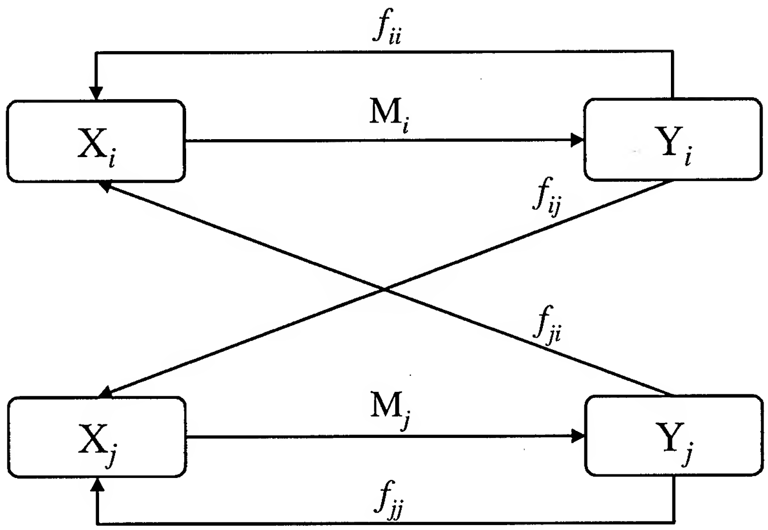

Figure 7: Generalized Relationships between Models

Input/Output Partitions. It is convenient for clarity to consider X_i and Y_i as being comprised of three partitions (each). X_i consists of one partition that can be compared to X_j in order to determine that $x_i = x_j$ and, therefore that two models are modeling the same scenario. $X_i \cap X_j$ effectively determines this partition of X_i (although it is doubtful in practice that every opportunity to compare values between the models will be taken advantage of, depending on the size and complexity of the input sets). There is another partition of X_i that consists of the information taken from another model to be used in model i. $X_i \cap f_{ji}(Y_j)$ (that is, the non-trivial portion of $f_{ji}(Y_j)$) determines this part of X_i, and I have labeled this partition previously as X_{ji}. The elements of the final partition of

X_i are simply every variable or parameter in X_i that is not a member of the other two partitions.

Y_i is similarly partitioned. The first partition of Y_i is comprised of that information which may be compared to Y_j in a covalidation effort. $Y_i \cap Y_j$ determines this partition. The second partition consists of the information in Y_i used to feedback to model *j*. This information is found in the non-trivial portion of $f_{ij}(Y_i) \cap X_j$, which I have previously labeled as Y_{ij}. Finally, the information not used for comparison to another model or for feedback comprises the final partition in Y_i. While it is not prohibited for a variable to belong to more than one partition of X_i or Y_i, it is generally the case that the three partitions are mutually exclusive and collectively exhaustive partitions for X_i and Y_i.

If partitions are constructed for *m* models ($m > 2$), the union of each pair-wise partition may be considered. For instance, the first partition in X_i consists of $(X_i \cap X_1) \cup \ldots \cup (X_i \cap X_{i-1}) \cup (X_i \cap X_{i+1}) \cup \ldots \cup (X_i \cap X_m)$ (omitting the trivial possibility of comparison to itself). Similarly, the second (feedback) partition consists of $(X_i \cap f_{1i}(Y_1)) \cup \ldots \cup (X_i \cap f_{mi}(Y_m))$ (allowing the possibility for feedback to itself). In Y_i, the first partition consists of $(Y_i \cap Y_1) \cup \ldots \cup (Y_i \cap Y_{i-1}) \cup (Y_i \cap Y_{i+1}) \cup \ldots \cup (Y_i \cap Y_m)$ (again omitting the possibility of comparison to itself). The feedback partition consists of $(f_{i1}(Y_i) \cap X_1) \cup \ldots \cup (f_{im}(Y_i) \cap X_m)$ (allowing the possibility for feedback to itself). The partitioning of a system of more than two models makes it likely that the partitions are not mutually exclusive and collectively exhaustive. Consider two models, each of which outputs a certain parameter. The parameter could be fed back from one or both models to a third model and the parameter could also be a point of comparison between the two models from which it is output.

Defining the Crossflow Mathematically. We wish to define a composition which maps a vector of m model inputs, $X_1,\ldots,X_m$ (each of which is an input vector to one of m models), back onto itself. Further, this mapping will account for the m models acting upon the inputs and for any feedback desired between the models. The vector of model inputs $X_1,\ldots,X_m$ I call $\tilde{\mathbf{X}}$ (i.e., $\tilde{\mathbf{X}} = X_1,\ldots,X_m)^T$, as shown in Equation (5).

$$\tilde{\mathbf{X}} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} \tag{5}$$

Similarly, I define a vector of model outputs $Y_1,\ldots,Y_m$ called $\tilde{\mathbf{Y}}$ (i.e., $\tilde{\mathbf{Y}} = Y_1,\ldots,Y_m)^T$, as shown in Equation (6).

$$\tilde{\mathbf{Y}} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{bmatrix} \tag{6}$$

As M_i maps X_i to Y_i, I construct a diagonal mapping matrix of models $M_1,\ldots,M_m$, designated $\tilde{\mathbf{M}}$, which maps $\tilde{\mathbf{X}}$ into $\tilde{\mathbf{Y}}$. This diagonal mapping matrix is constructed by entering M_i as the i^{th} diagonal element on the mapping matrix. All other elements of the mapping matrix are “0”, as described by Equation (7).

$$\tilde{\mathbf{M}} = \begin{bmatrix} \mathrm{M}_1 & 0 & \cdots & 0 \\ 0 & \mathrm{M}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{M}_m \end{bmatrix} \tag{7}$$

The matrix mapping of *m* models working on their respective inputs can be written as in Equation (8).

$$\begin{aligned} \tilde{\mathbf{M}}(\tilde{\mathbf{x}}) &= \begin{bmatrix} \mathrm{M}_1 & 0 & \cdots & 0 \\ 0 & \mathrm{M}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{M}_m \end{bmatrix} \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \\ \mathrm{x}_m \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{M}_1(\mathrm{x}_1) \\ \mathrm{M}_2(\mathrm{x}_2) \\ \vdots \\ \mathrm{M}_m(\mathrm{x}_m) \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{y}_1 \\ \mathrm{y}_2 \\ \vdots \\ \mathrm{y}_m \end{bmatrix} \\ &= \tilde{\mathbf{y}} \end{aligned} \tag{8}$$

where $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$ and $\tilde{\mathbf{y}} \in \tilde{\mathbf{Y}}$.

An m by m mapping matrix of feedback functions, denoted by $\tilde{f}$, is also constructed and shown in Equation (9). In this matrix, entry (i, j) contains the feedback function f_{ji}. For all f_{ji}, $j \neq i$, the feedback is the change to be applied to the input of the i^{th} model based on the output of the j^{th} model. In other words, f_{ji}, $j \neq i$, returns a "0" for all entries except the variables or parameters requiring changes. For these, the difference between the original value of the variable or parameter and the desired value is returned. The feedback f_{ii} is constructed as the input (from the current iteration) to the i^{th} model with any "feedback to self" variables or parameters modified accordingly.

$$\tilde{f} = \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{m1} \\ f_{12} & f_{22} & \cdots & f_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1m} & f_{2m} & \cdots & f_{mm} \end{bmatrix} \tag{9}$$

In this way, the i^{th} row contains both the input to model i as well as the desired changes to be applied to this input. The vector is determined by the vector sum of each model's feedback to model i (including f_{ii}). As demonstrated in Equation (4), when the i^{th} row is applied to the m model outputs, the result is x_i. Note that this construction allows the possibility that more than one model could influence a single parameter in model i. If this is the case, feedback values may be appropriately weighted (equally weighted to reflect an average, perhaps) or some other method may be employed to resolve the multiple feedback, but here I assume f_{ji} is constructed such that only one model is allowed to affect any one element of the x_i vector. The vector consisting of the changes made to all m models, $(x_1, x_2,\ldots, x_m)^T$, has already been labeled $\tilde{\mathbf{x}}$. The mapping matrix $\tilde{f}$, then, maps $\tilde{\mathbf{Y}}$ into $\tilde{\mathbf{X}}$ as follows:

$$\begin{aligned}
\tilde{f}(\tilde{\mathbf{y}}) &= \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{m1} \\ f_{12} & f_{22} & \cdots & f_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1m} & f_{2m} & \cdots & f_{mm} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \\
&= \begin{bmatrix} f_{11}(y_1) \oplus f_{21}(y_2) \oplus \ldots \oplus f_{m1}(y_m) \\ f_{12}(y_1) \oplus f_{22}(y_2) \oplus \ldots \oplus f_{m2}(y_m) \\ \vdots \\ f_{1m}(y_1) \oplus f_{2m}(y_2) \oplus \ldots \oplus f_{mm}(y_m) \end{bmatrix} \\
&= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \\
&= \tilde{\mathbf{x}}
\end{aligned} \tag{10}$$

The desired composition of mappings from $\tilde{\mathbf{X}}$ back to itself, then, is merely a function of $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$ that I will call $\tilde{\mathbf{F}}$. The construction of this function of $\tilde{\mathbf{x}}$ is shown in Equation (11).

$$\begin{aligned}
\tilde{\mathbf{F}}(\tilde{\mathbf{x}}) &= \tilde{f}\left(\tilde{\mathbf{M}}(\tilde{\mathbf{x}})\right) \\
&= \tilde{f}(\tilde{\mathbf{y}}) \\
&= \tilde{\mathbf{x}}
\end{aligned} \tag{11}$$

The complete expansion of $\tilde{\mathbf{F}}$ is demonstrated in Equation (12).

$$
\begin{aligned}
\tilde{\mathbf{F}}(\tilde{\mathbf{x}}) &= \tilde{\mathbf{F}}\begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \\ \mathrm{x}_m \end{bmatrix} \\
&= \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{m1} \\ f_{12} & f_{22} & \cdots & f_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1m} & f_{2m} & \cdots & f_{mm} \end{bmatrix}
\begin{bmatrix} \mathrm{M}_1 & 0 & \cdots & 0 \\ 0 & \mathrm{M}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathrm{M}_m \end{bmatrix}
\begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \\ \mathrm{x}_m \end{bmatrix} \\
&= \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{m1} \\ f_{12} & f_{22} & \cdots & f_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ f_{1m} & f_{2m} & \cdots & f_{mm} \end{bmatrix}
\begin{bmatrix} \mathrm{M}_1(\mathrm{x}_1) \\ \mathrm{M}_2(\mathrm{x}_2) \\ \vdots \\ \mathrm{M}_m(\mathrm{x}_m) \end{bmatrix} \\
&= \begin{bmatrix} f_{11}(\mathrm{M}_1(\mathrm{x}_1)) \oplus f_{21}(\mathrm{M}_2(\mathrm{x}_2)) \oplus \ldots \oplus f_{m1}(\mathrm{M}_m(\mathrm{x}_m)) \\ f_{12}(\mathrm{M}_1(\mathrm{x}_1)) \oplus f_{22}(\mathrm{M}_2(\mathrm{x}_2)) \oplus \ldots \oplus f_{m2}(\mathrm{M}_m(\mathrm{x}_m)) \\ \vdots \\ f_{1m}(\mathrm{M}_1(\mathrm{x}_1)) \oplus f_{2m}(\mathrm{M}_2(\mathrm{x}_2)) \oplus \ldots \oplus f_{mm}(\mathrm{M}_m(\mathrm{x}_m)) \end{bmatrix} \\
&= \begin{bmatrix} f_{11}(\mathrm{y}_1) \oplus f_{21}(\mathrm{y}_2) \oplus \ldots \oplus f_{m1}(\mathrm{y}_m) \\ f_{12}(\mathrm{y}_1) \oplus f_{22}(\mathrm{y}_2) \oplus \ldots \oplus f_{m2}(\mathrm{y}_m) \\ \vdots \\ f_{1m}(\mathrm{y}_1) \oplus f_{2m}(\mathrm{y}_2) \oplus \ldots \oplus f_{mm}(\mathrm{y}_m) \end{bmatrix} \\
&= \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \\ \vdots \\ \mathrm{x}_m \end{bmatrix} \\
&= \tilde{\mathbf{x}}
\end{aligned}
\tag{12}
$$

This composition accepts each model's initial input, executes each model once, and delivers feedback to each model's input based on the model runs. The result is an updated set of model inputs which, assuming the validity of the feedback functions, provide better estimates for some of the input values than the initial set of approximated inputs. Successive applications of $\tilde{\mathbf{F}}$ are denoted by superscript, as are the iteration numbers on an element of $\tilde{\mathbf{X}}$ (whose initial value for a particular scenario is assumed to be $\tilde{\mathbf{x}}^0 \in \tilde{\mathbf{X}}$), as illustrated in Equation (13).

$$\begin{aligned} \tilde{\mathbf{x}}^n &= \tilde{\mathbf{F}}\left(\tilde{\mathbf{x}}^{n-1}\right) \\ &= \tilde{\mathbf{F}}^2\left(\tilde{\mathbf{x}}^{n-2}\right) \\ &= \tilde{\mathbf{F}}\left(\tilde{\mathbf{F}}^{n-1}\left(\tilde{\mathbf{x}}^0\right)\right) \\ &= \tilde{\mathbf{F}}^n\left(\tilde{\mathbf{x}}^0\right) \end{aligned} \tag{13}$$

Since we desire successive iterates to approach some "truth" or "best" value, we would like to know that such a value exists, under what conditions we can actually find it, and lastly, how to find it. In order to address these issues, we need to make some assumptions concerning $\tilde{\mathbf{X}}$, $\tilde{\mathbf{M}}$, and $\tilde{f}$. Recall that $\tilde{\mathbf{X}}$ is the vector of model input sets $X_1,\ldots,X_m$. Each of these X_i is a set in $\mathbf{R}^{n_i}$ where $1 \leq n_i < \infty$. Further, we assume each X_i is closed and bounded, or compact. This implies that $\tilde{\mathbf{X}}$ is also compact for $n = \sum n_i < \infty$. Assume that the composition $\tilde{f} \circ \tilde{\mathbf{M}}$ is continuous. Though this assumption is not true in general and may seem questionable, recall that $\tilde{f}$ only returns a change to inputs whose initial estimates are in question. All other inputs to the composition are continuous, since they do not change at all. The estimated inputs, then, are real-valued variables such as arrival and service rates, or efficiency factors. Given that the variables affecting $\tilde{f}$ are real-valued, the assumption of the composition's continuity is realistic. Given these conditions, it can be stated by Brouwer's fixed-point theorem that $\tilde{\mathbf{F}}$ has a fixed point in $\tilde{\mathbf{X}}$ (Border 1985). Of note is that while there is at least one fixed point in $\tilde{\mathbf{X}}$, the stated assumptions are not enough to assert that the fixed point is unique. If, however, the composition $\tilde{\mathbf{F}}$ is a contraction, it can be proved that there is only one fixed point.

Definition: Fixed Point and Contraction. Let $\tilde{\mathbf{F}}: \tilde{\mathcal{X}} \rightarrow \tilde{\mathcal{X}}$ be a function from a metric space ($\tilde{\mathcal{X}}$, $d_{\tilde{\mathcal{X}}}$) into itself. A point $\tilde{x}^*$ is called a fixed point of $\tilde{\mathcal{X}}$ if $\tilde{\mathbf{F}}\left(\tilde{x}^*\right)=\tilde{x}^*$. The function $\tilde{\mathbf{F}}$ is called a contraction of $\tilde{\mathcal{X}}$ if there is a positive number $\alpha < 1$, such that

$$d_{\tilde{\mathcal{X}}}\left(\tilde{\mathbf{F}}\left(\tilde{x}^i\right),\tilde{\mathbf{F}}\left(\tilde{x}^j\right)\right)\leq \alpha d_{\tilde{\mathcal{X}}}\left(\tilde{x}^i,\tilde{x}^j\right) \text{ for all } \tilde{x}^i,\tilde{x}^j \text{ in } \tilde{\mathcal{X}} \tag{14}$$

(Apostol, 1974). In other words, this inequality requires that the value of the distance between the images under $\tilde{\mathbf{F}}$ of any two elements of $\tilde{\mathcal{X}}$ be less than the value of the distance between the elements.

Proof of Feedback Convergence. Theorem: If the composition $\tilde{\mathbf{F}} = \tilde{f} \circ \tilde{\mathbf{M}}$ is a contraction of the complete metric space ($\tilde{\mathcal{X}}$, $d_{\tilde{\mathcal{X}}}$), then $\tilde{\mathbf{F}}$ has a unique fixed point in $\tilde{\mathcal{X}}$.

Let d be a metric on $\tilde{\mathcal{X}}$ (a set of $\tilde{\mathbf{X}}$), such that ($\tilde{\mathcal{X}}$, $d_{\tilde{\mathcal{X}}}$) is a metric space. Now, each $\mathcal{X}_i$ is an n_i-dimensional vector on which some metric $d_{\mathcal{X}_i}$ (the Euclidean metric is a reasonable choice) can be assumed to exist, and therefore, each ($\mathcal{X}_i, d_{\mathcal{X}_i}$) is a metric space. I now design a metric $d_{\tilde{\mathcal{X}}}$ for $\tilde{\mathcal{X}}$ such that I can determine the value of the metric for each of the m $\mathcal{X}_i$'s and determine the value of $d_{\tilde{\mathcal{X}}}$ for the resulting m-dimensional vector. The choice of metrics is arbitrary, but the Euclidean distance for m-dimensional vectors or an Euclidean metric determined after applying positive, non-zero weights based on the relative importance of each model is valid and seems reasonable. Since this metric defined on $\tilde{\mathcal{X}}$ is a metric developed by the combination of m other (valid) metrics, it is clear that the resulting metric is itself a metric, and $(\tilde{\mathcal{X}}, d_{\tilde{\mathcal{X}}})$ is a metric space. Further, since $\tilde{\mathcal{X}}$ has already been assumed to be a compact set, $(\tilde{\mathcal{X}}, d_{\tilde{\mathcal{X}}})$ is also a complete metric space. Given that $(\tilde{\mathcal{X}}, d_{\tilde{\mathcal{X}}})$ is a complete metric space, if it can be asserted that $\tilde{\mathbf{F}} = \tilde{f} \circ \tilde{\mathbf{M}}$ is a contraction mapping on $\tilde{\mathcal{X}}$, I will show that $\tilde{\mathbf{F}}$ has a unique fixed point in $\tilde{\mathcal{X}}$ (Apostol 1974). Further, this fixed point is invariant regardless of the metric used, and an intuitive and simple method exists for determining the fixed point in $\tilde{\mathcal{X}}$.

However, determining that $\tilde{\mathbf{F}}$ is indeed a contraction mapping is likely not a trivial task, given the complex nature of mathematical models. Decomposing $\tilde{\mathbf{F}}$ into $\tilde{f}$ and $\tilde{\mathbf{M}}$ may be helpful for determining whether $\tilde{\mathbf{F}}$ is a contraction or not. I consider from the output of $\tilde{\mathbf{M}}$ that partition of each $\mathcal{Y}_i$ that is used for feedback to other models. Further, I assume that $\tilde{\mathbf{M}}$ is Lipschitz continuous across this partition. (Since the other partitions of the $\mathcal{Y}_i$'s are not used by $\tilde{f}$, Lipschitz continuity is not established across these partitions. Despite the limited mapping (from $\mathcal{X}_i$ to a partition of $\mathcal{Y}_i$) considered, I continue to use $\tilde{\mathbf{M}}$ as its description.)

Definition: Lipschitz Continuity. A function $\tilde{\mathbf{F}}$ is said to be Lipschitz continuous at a point $\tilde{x}^i$ if there exists β > 0, L > 0, and a unit hypersphere $B(\tilde{x}^i)$ such that

$$\left\|\tilde{\mathbf{F}}\left(\tilde{x}^j\right)-\tilde{\mathbf{F}}\left(\tilde{x}^i\right)\right\| < \mathrm{L}\left\|\tilde{x}^j-\tilde{x}^i\right\|^{\beta} \tag{15}$$

whenever $\tilde{x}^j \in B\left(\tilde{x}^i\right), \tilde{x}^j \neq \tilde{x}^i$. Notice that this condition is a relaxed form of the condition required for a function to be a contraction. The general application of the Lipschitz condition is to ensure the existence of a derivative of the function at $\tilde{x}^i$ if $\beta > 1$ (Apostol 1974).

This implies that there exists $0 \leq \mathrm{L}_1 < \infty$ such that the following condition holds.

$$d_{\tilde{\mathcal{Y}}}\left(\tilde{\mathbf{M}}\left(\tilde{x}^i\right), \tilde{\mathbf{M}}\left(\tilde{x}^j\right)\right) \leq \mathrm{L}_1 d_{\tilde{\mathcal{X}}}\left(\tilde{x}^i, \tilde{x}^j\right) \text{ for all } \tilde{x}^i, \tilde{x}^j \text{ in } \tilde{\mathcal{X}} \tag{16}$$

Now I can show that there exists some non-trivial $\tilde{f}$ such that $\tilde{\mathbf{F}}$ must be a contraction. First, $\tilde{f}$ must also be Lipschitz continuous. This property implies that there exists $0 \leq \mathrm{L}_2 < \infty$ such that the inequality in Equation (17) holds.

$$d_{\tilde{\mathcal{X}}}\left(\tilde{f}\left(\tilde{y}^i\right), \tilde{f}\left(\tilde{y}^j\right)\right) \leq \mathrm{L}_2 d_{\tilde{\mathcal{Y}}}\left(\tilde{y}^i, \tilde{y}^j\right) \text{ for all } \tilde{y}^i, \tilde{y}^j \text{ in } \tilde{\mathcal{Y}} \tag{17}$$

If I combine the implications based on the assumptions of Lipschitz continuity, Equation (18) is obtained.

$$\begin{aligned} d_{\tilde{\mathcal{X}}}\left(\tilde{f}\left(\tilde{y}^{i}\right), \tilde{f}\left(\tilde{y}^{j}\right)\right) &\leq \mathrm{L}_2 d_{\tilde{y}}\left(\tilde{y}^{i}, \tilde{y}^{j}\right) \\ &= \mathrm{L}_2 d_{\tilde{y}}\left(\tilde{\mathbf{M}}\left(\tilde{x}^{i}\right), \tilde{\mathbf{M}}\left(\tilde{x}^{j}\right)\right) \\ &\leq \mathrm{L}_2 \mathrm{L}_1 d_{\tilde{\mathcal{X}}}\left(\tilde{x}^{i}, \tilde{x}^{j}\right) \end{aligned} \tag{18}$$

Since $\tilde{f}$ is constructed by the modeler, it is clear that a non-trivial $\tilde{f}$ may be specified which forces the product $\mathrm{L}_2\mathrm{L}_1 < 1$. In this manner, the composition $\tilde{\mathbf{F}} = \tilde{f} \circ \tilde{\mathbf{M}}$ may be forced to be a contraction mapping on $\tilde{\mathcal{X}}$ with $\alpha = \mathrm{L}_2\mathrm{L}_1 < 1$.

The property of contraction in $\tilde{\mathbf{F}}$ allows the use of an iterative method for determining the fixed point of $\tilde{\mathbf{F}}$ in $\tilde{\mathcal{X}}$. Initially, some $\tilde{x}^0 \in \tilde{\mathcal{X}}$ is selected. While it seems logical to select an $\tilde{x}^0$ thought to be close to the fixed point, the selection of $\tilde{x}^0$ does not determine whether or not the fixed point will be found, but could affect the number of iterations required to find it. With an initial $\tilde{x}^0$, $\tilde{\mathbf{M}}$, the diagonal mapping matrix of models, is performed (i.e., each model is run using their respective $x_i^0 \in \mathcal{X}_i$, i = 1,..., m). Next, the feedback matrix $\tilde{f}$ is implemented, and the result is $\tilde{x}^1$. This cycle from $\tilde{x}^0$ to $\tilde{x}^1$ is the first iteration. (Each iteration may alter only those inputs which are affected by the feedback functions, all other inputs values remain fixed.) Repeating the process during a second iteration yields $\tilde{x}^2$. From the assumption that $\tilde{\mathbf{F}}$ is a contraction of $\tilde{\mathcal{X}}$, it follows that $\tilde{x}^1$ and $\tilde{x}^2$ are closer to each other (in terms of the selected metric) than $\tilde{x}^0$ and $\tilde{x}^1$. I define a sequence $\{\tilde{x}^n\}$ of iterates as the sequence given by

$$\tilde{\mathbf{x}}^0,\ \tilde{\mathbf{x}}^1 = \tilde{\mathbf{F}}\left(\tilde{\mathbf{x}}^0\right),\ \tilde{\mathbf{x}}^2 = \tilde{\mathbf{F}}^2\left(\tilde{\mathbf{x}}^0\right) = \tilde{\mathbf{F}}\left(\tilde{\mathbf{F}}\left(\tilde{\mathbf{x}}^0\right)\right), \ldots,\ \tilde{\mathbf{x}}^n = \tilde{\mathbf{F}}^n\left(\tilde{\mathbf{x}}^0\right) \tag{19}$$

An implication of the n^{th} iteration is given in Equation (20).

$$\begin{aligned} d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{x}}^n, \tilde{\mathbf{x}}^{n-1}\right) &= d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{F}}\left(\tilde{\mathbf{x}}^{n-1}\right), \tilde{\mathbf{F}}\left(\tilde{\mathbf{x}}^{n-2}\right)\right) \\ &\le \alpha d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{x}}^{n-1}, \tilde{\mathbf{x}}^{n-2}\right) \end{aligned} \tag{20}$$

Inductively, I arrive at Equation (21).

$$\begin{aligned} d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{x}}^n, \tilde{\mathbf{x}}^{n-1}\right) &\le \alpha^{n-1} d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^0\right) \\ &= c\alpha^{n-1} \end{aligned} \tag{21}$$

In Equation (21), $c = d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^0\right)$. Applying the triangle inequality for $m > n$, Equation (22) is obtained.

$$\begin{aligned} d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{x}}^m, \tilde{\mathbf{x}}^n\right) &\le \sum_{k=n}^{m-1} d_{\tilde{\mathfrak{X}}}\left(\tilde{\mathbf{x}}^{k+1}, \tilde{\mathbf{x}}^k\right) \\ &\le c\sum_{k=n}^{m-1} \alpha^k \\ &= c\frac{\alpha^n - \alpha^m}{1-\alpha} \\ &< \frac{c}{1-\alpha}\alpha^n \end{aligned} \tag{22}$$

Since α < 1, $\alpha^n \to 0$ as $n \to \infty$. This implies that $\{\tilde{x}^n\}$ is a Cauchy sequence in ($\tilde{\mathcal{X}}$, $d_{\tilde{\mathcal{X}}}$). However, $\tilde{\mathcal{X}}$ is a complete metric space, so there exists a point $\tilde{x}^* \in \tilde{\mathcal{X}}$ such that $\tilde{x}^n \to \tilde{x}^*$. Under the assumption of the continuity of $\tilde{\mathbf{F}}$, I find the fixed point by taking the limit of the sequence $\{\tilde{x}^n\}$.

$$\begin{aligned}\tilde{\mathbf{F}}(\tilde{x}^*) &= \tilde{\mathbf{F}}\left(\lim_{n\to\infty} \tilde{x}^n\right)\\ &= \lim_{n\to\infty} \tilde{\mathbf{F}}(\tilde{x}^n)\\ &= \lim_{n\to\infty} \tilde{x}^{n+1}\\ &= \tilde{x}^*\end{aligned} \tag{23}$$

Therefore, $\tilde{x}^*$ is the fixed point of $\tilde{\mathbf{F}}$. Further, finding the fixed point is simply a matter of performing repeated applications of the mapping $\tilde{\mathbf{F}}$ until satisfactory stopping criteria have been satisfied. Ultimately, this criteria should be that the models' inputs have converged at $\tilde{x}^*$.

If absolute convergence does not occur in a reasonable number of iterations, however, there are several possible explanations and courses of action. The most discouraging of these is that the $\tilde{\mathbf{F}}$ mapping is not actually a contraction mapping. This phenomenon can be detected if the difference between successive input values grows instead of decreases. Related to this problem, the $\tilde{\mathbf{F}}$ mapping could be only marginally a contraction (i.e., α from Equation (14) very close to 1.0), and successive iterates become very marginally closer. In either of these cases, a new $\tilde{f}$ may be sought so the composition $\tilde{\mathbf{F}}$ is a (stronger) contraction mapping.

Another problem that may prevent convergence to $\tilde{x}^*$ is that successive iterations indicate that a point of diminishing returns has been reached. This case may be manifested in one of two ways. First, iterates may steadily approach the fixed point, but the high fidelity of model inputs allows very incremental changes to the input values, so the actual fixed point may be attainable only through very many iterations. Another possibility is that limitations of the models prohibit the use of the actual fixed point as an input, and the method may alternate between points near the actual fixed point. In either of these cases, the use of the last input values is acceptable as a proxy for the actual $\tilde{x}^*$. If the $\tilde{\mathbf{F}}$ mapping is indeed a contraction mapping, I can be confident that this point is very close to the actual fixed point.

It is an interesting note that convergence occurs not only in the model inputs, $\tilde{\mathcal{X}}$, but also in the model outputs, $\tilde{\mathcal{Y}}$. This is easily demonstrated by first performing $\tilde{\mathbf{F}}$ on the fixed point, $\tilde{x}^* \in \tilde{\mathcal{X}}$. Of course, the result is $\tilde{x}^*$. However, if one performs only the model portion of the $\tilde{\mathbf{F}} = \tilde{f} \circ \tilde{\mathbf{M}}$ composition on $\tilde{x}^*$, a fixed output is obtained, as well, which I denote by $\tilde{y}^*$.

$$\tilde{\mathbf{M}}\left(\tilde{x}^*\right) = \tilde{y}^* \tag{24}$$

Gradient Analysis

In this section I develop a means for assessing the "closeness" of the models. At the conclusion of the iterative method, we have fixed points from each model that may be compared. Rather than simply relying on this comparison made at a point, I propose estimating the gradient of each model relative to selected inputs and comparing these vectors. Metamodels constructed across a small experimental design of relevant input variables provide a convenient means of effecting this comparison. Both models are executed (during the iterative scheme) at the same basic input settings. The final input settings from the iterative scheme become the center point of an experimental design aimed at estimating the local gradients of each model. A comparison of these gradient estimates indicates whether or not the models respond in a similar fashion to perturbations in the selected inputs.

The method I develop allows comparison between both the relative closeness of average model outputs and estimates of gradients of the models representing the sensitivity of a selected output to a set of common inputs. Using this method, I am able to investigate both the relative predictive values of the metamodels (through the output comparison) as well as compare the metamodels' abilities to provide description of the physical system (through comparison of local gradients). The method is summarized in the following subsections. The first describes the creation of an experimental design, and the second describes the comparison of the resulting metamodels.

Experimental Design. The experimental design chosen for this study is based on the desire to evaluate the sensitivities of a single output to changes in key inputs. The choice of design depends on the study. The number of model executions to be performed is limited by the size of the models, the number of input factors to be varied, the desired design resolution, and the number of replications to be made at each design point (to reduce the output variability of simulations) (Box and Draper 1987).

The result of the iterative scheme is a fixed point of the models' inputs. This fixed point provides the input set to each model (to include parameters and variables) which corresponds as nearly as possible to the inputs and outputs (as applicable) of the opposing model. This input establishes the center point for the experimental design. It is significant that the convergence that results from the iterative scheme is valid only for the design point upon which the scheme was performed. I am assuming that the perturbations used in the experimental design are small enough so that the effect of using the same fixed points across the entire design is insignificant.

Since exercising the technique across a wide experimental design may yield convergence to greatly different sets of specific model inputs, the iterative method would require repetition at each design point. For instance, a large change in the number of available aircraft (or any resource) may significantly affect optimal routing (one of the crossflow variables). For this reason and since one of the goals is gradient approximation at the center design point, I assume that design points are very close to one another (say, approximately plus or minus ten percent) and that the potential differences in fixed points are unimportant. The desired closeness of the design points creates a trade-off with the number of runs required for a model with output variability (such as a simulation) so that a statistically significant difference may be seen between the design points.

Model Comparison. I recognize there there are a number of methods available for comparing the result of the experimental design. This research describes two methods. The first method is demonstrated on the illustrative example of the next section. This method considers the fact that, if two models are the same, their outputs for a given set of inputs, should equal each other. Therefore, the difference between the output sets should be zero. Further, the gradients of an output of interest with respect to the inputs of interest should be the same, also. In other words, the way in which an input affects one model should be the same way it affects the other model, if the models are covalid. This implies that the difference between the gradients of interest of the models should also be zero (for two identical models).

The proposal, then, is to take the (design) point-by-(design) point difference between the outputs of each model. In this application, I compare a simulation model in which several runs are performed at each design point with an optimization model where only one run is required at each design point. I take the difference between each simulation run and the corresponding optimization run, since if I operate the optimization multiple times at the point, the same result would be obtained. A regression model created from these difference points should be a "zero" model, with only random variation providing any deviation from zero. Not only should the mean equal zero, but the coefficients should all be zero, as well. The analysis of the regression statistics, then, provides the extent to which the two models are covalid. Of course, the specific application determines how close to zero the regression model has to be to deem the models covalid, but the significance of the model, the significance of the coefficients, and the amount of variation explained by the model (R^2) are factors that can help in making the determination.

The other method I employ takes advantage of the difference between optimization and the simulation model. This method considers that the resulting gradient direction found by each model should be similar. As such, the angle between the gradient vectors created by each metamodel should be relatively small.

Box and Draper describe the computation of a confidence region around the coefficients of a linear regression that I employ here. A confidence region shaped as a hypercone is constructed about the gradient vector of the simulation metamodel (chosen since there will likely be more variability in this metamodel). The following equation is used to determine the angle about the determined gradient vector the confidence region extends (Box and Draper 1987).

$$\sin\theta = \left\{ \frac{(k-1)s_b^2 F_\alpha(k-1, \nu_b)}{\sum_{i=1}^{k} b_i^2} \right\}^{1/2} \tag{25}$$

In the equation, θ is the angle the confidence region extends from the estimated gradient vector, k is the number of variables estimated, b is the set of coefficients, s_b is the standard error of each coefficient, and ν_b is the number of degrees of freedom on which s_b is based.

If the gradient vector of the optimization model lies in the constructed hypercone, we may not declare that a difference exists between the two gradient vectors. The closeness of the mean values of the metamodels should also be considered, however, before making a determination as to the two models covalidity.

Illustrative Example

In order to demonstrate the use of the proposed methodology, an example is given using very small-scale test models that represent the MASS and NRMO models. Data used in these small-scale surrogates are notional only and no inferences should be made from the data or results to either the MASS and NRMO models or to any actual airlift scenario.

The Baby Models. The scenario posed is that 50 similar aircraft must fly as many missions as possible from a home base to either of two air bases (A and B) in 15 days. The air bases can handle 10 and 5 aircraft maximum on the ground (MOG) at a time, respectively, and are different distances from the home base. Sensitivities of both the number of aircraft and the amount of available MOG are of interest here. In the simulation (Baby MASS), the flight times to (and from) each base and the ground time of the aircraft at bases A and B are random variables, while the optimization (Baby NRMO) assumes a mean value. (See Appendix A for a more detailed account of both the simulation and optimization models.)

The simulation accepts as input a proportion of use for the two bases (the percentage of missions flown to each base), while the optimization yields optimal values for this proportion. Similarly, since it is not possible in practice to optimally schedule the ground spaces available at the bases (as an optimization model would), the optimization accepts as input a MOG efficiency factor. The simulation may yield a practical maximum number of aircraft that can be serviced at a base from which a better estimate for MOG efficiency may be derived. The iterative scheme uses this information to attempt convergence to some "optimal" proportion of base visitation and MOG efficiency.

Output/Input Crossflow Method. Table 1 shows the results of the iterative scheme with 60 runs being made at each iteration (for the simulation). The table displays two rows of information for each iteration. In the Baby MASS column, the top row of each iteration shows the percentage of aircraft sent to each base, while the bottom row displays the average number of missions flown to each base, as well as the total average number of missions flown over the 15 day period. In the Baby NRMO column, the top row shows the MOG efficiency applied to Base B (the bottlenecked base), while the bottom row again shows the number of missions flown to each base, as well as the total number of missions flown.

A fifty-fifty split was used as the nominal value for the proportion of aircraft sent to each base, and 1.0 was used as the starting MOG efficiency. When the simulation results indicated a bottleneck at a base, a MOG efficiency was calculated based on the number of aircraft which were actually able to be serviced at the base and the amount of service time the aircraft had at the base (see Appendix A). For a Baby NRMO run, the number of planes routed to each base is determined. This proportion is used as direct input for the subsequent Baby MASS run (see Appendix A).

Table 1: Iterative Scheme for Test Models

i	Baby MASS			Baby NRMO		
	A	B	Total	A	B	Total
1	0.5	0.5	← %	efficiency →	1.0	
	87.2	92.5	179.7	52	140	192
2	0.2857	0.7143	← %	efficiency →	0.9479	
	51.1	130.3	181.4	56.1	132.7	188.8
3	0.3127	0.6873	← %	efficiency →	0.9337	
	55.8	128.4	184.2	57.2	130.7	187.9
4	0.3202	0.6798	← %	efficiency →	0.9234	
	57.0	127.0	184.0	58.0	129.3	187.3
5	0.3257	0.6743	← %	efficiency →	0.9212	
	57.7	126.7	184.4	58.1	129.0	187.1
6	0.3269	0.6731	← %	efficiency →	0.9172	
	57.7	126.1	183.8	58.5	128.4	186.9
7	0.3290	0.6710	← %	efficiency →	0.9063	
	58.4	124.6	183.0	59.3	126.9	186.2
8	0.3348	0.6652	← %	efficiency →	0.9126	
	60.0	125.5	185.5	58.8	127.8	186.6
9	0.3314	0.6686	← %	efficiency →	0.9099	
	58.5	125.1	183.6	59.0	127.4	186.4
10	0.3329	0.6671	← %	efficiency →	0.9098	
	59.3	125.1	184.4	59.0	127.4	186.4
11	0.3329	0.6671	← %	efficiency →	0.9098	
	59.3	125.1	184.4	59.0	127.4	186.4

As seen in Table 1, a stopping criterion is met since the Baby NRMO results for the 10th and 11th iterations are identical, indicating convergence. This input value convergence is shown graphically in Figure 8. The final output values indicate that an average of 184.4 missions are flown in Baby MASS (with standard error of 2.21) and 186.4 missions are flown in Baby NRMO. The comparison of these output values across the iterations is shown in Figure 9. The converged base-use proportions and MOG efficiency are used throughout the experimental design for gradient estimation.

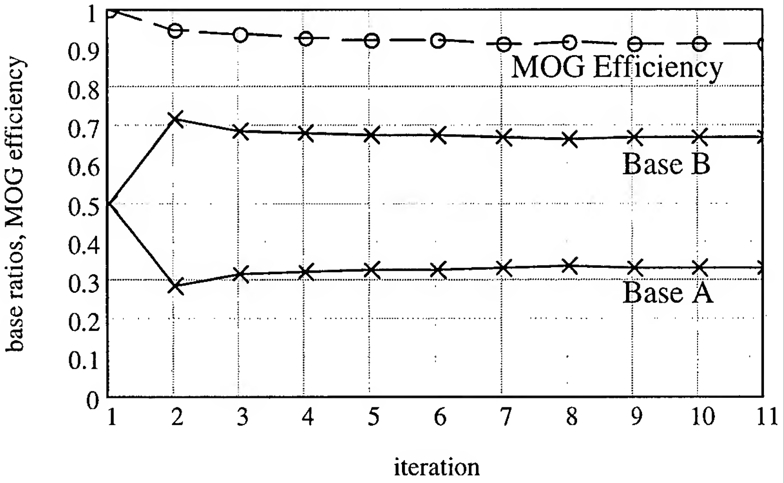

Figure 8: Input Convergence

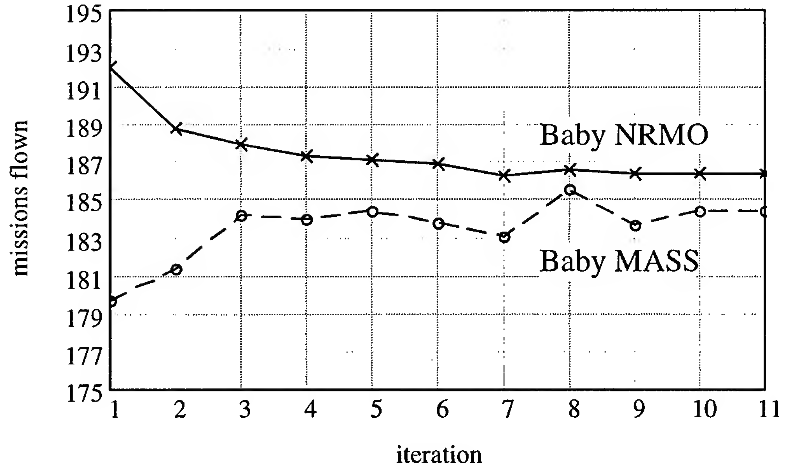

Figure 9: Throughput Convergence

Experimental Design. Metamodels are created using the number of aircraft and the amount of MOG at base B as independent variables perturbed over a 2^2 plus center point experimental design. (MOG at base B is selected since base B proved to be a system bottleneck.) The number of planes is varied plus and minus 10 percent (to 55 and 45 planes, respectively) and the MOG at base B is varied by plus and minus 1 unit of MOG (to 6 and 4 MOG units, respectively). The number of missions flown is the dependent variable.

The metamodel results are summarized in Table 2. Note that for the Baby NRMO model, the only error comes from specification bias. Since there is no random error in Baby NRMO's metamodel, it is clear that none of the design points are outside the critical region found at the design's center, i.e., the basis did not change at any of the design points. The metamodels are also shown graphically in Figures 10 and 11 with the number of missions flown shown on each of the vertical axes.

Table 2: Comparison of Test Model Results

		Baby MASS	Baby NRMO
Coefficient	Mean	174.6	186.4
	Planes	10.1	13.0
	MOG at B	15.6	11.3
	Interaction	5.6	0.0
P-Value	Mean	0.0	0.0
	Planes	0.0	0.0
	MOG at B	0.0	0.0
	Interaction	0.0	1.0
Coefficient Standard Error		0.65	0.00
SSR		90441.8	1184.7
SSE		30170.6	0.0
SST		120612.4	1184.7
F*		295.8	10073834
F* Significance		0.00	0.00
R^2		0.750	1.0

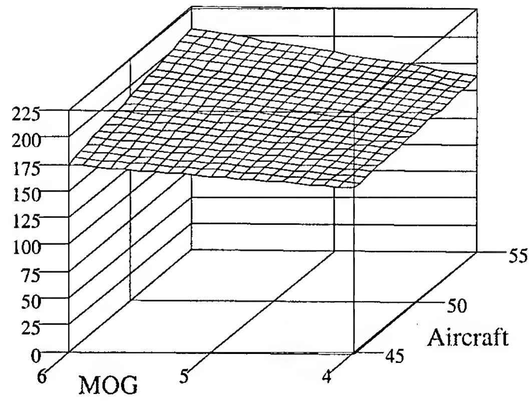

Figure 10: Baby MASS Metamodel

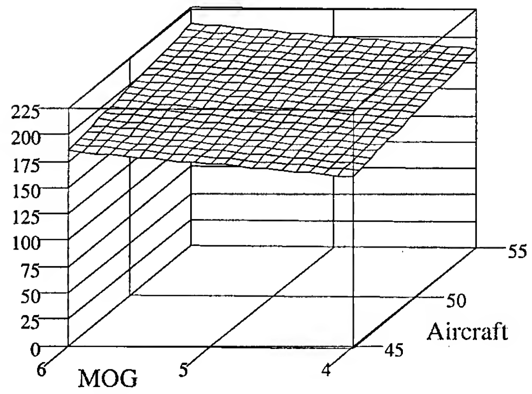

Figure 11: Baby NRMO Metamodel

I create a new set of dependent variable data (termed "difference" data) by taking the difference between the Baby NRMO and the Baby MASS data at each design point. I then construct a metamodel for this "difference" data, again using MOG at base B and the number of aircraft as the independent variables. In this instance, however, the dependent variable represents the difference in the number of missions between the Baby NRMO and the Baby MASS models (at each design point). The results of the "difference" model are given in Table 3.

Table 3: "Difference" Model Results

		Difference
Coefficient	Mean	11.8
	Planes	2.9
	MOG at B	-4.3
	Interaction	-5.6
P-Value	Mean	0.0
	Planes	0.0
	MOG at B	0.0
	Interaction	0.0
Coefficient Standard Error		0.65
SSR		14115.4
SSE		30178.9
SST		44294.3
F*		46.1
F* Significance		0.0
R^2		0.319

The significance of this "difference" model implies disparities between the two models, with a large amount of the discrepancy reflected in the interaction term. Also, there is a significant difference in mean values between the two models (compared to the center point arrived at through the iterative scheme). The relatively large "difference" model is mainly the result of the MASS and NRMO models being evaluated across a relatively wide experimental region. (The MOG at base B factor is set at five units of MOG plus or minus 20 percent.) In order to alleviate this problem, the fixed-point could be re-determined at each design point or the design space could be lessened (recall that I recommend a deviation from the center point of ten percent). Here, however, the design space could not be narrowed, since the MOG at base B factor cannot be fractionalized in the simulation.

Determination of Covalidation. To determine the models' covalidity, the preceding facts and statistics are considered, as well as the specific use of the models and how imperative it is that the models yield identical outputs.

From the "difference" model regression summarized in Table 3, it is seen that all the coefficients and the mean of the model are significant, which implies that the models do not represent the same reality and are therefore not covalid. From a more practical standpoint, however, we see the models' similarity in Figures 10 and 11 and from the regressions that, at least, the coefficients are all of the same sign and same order of magnitude. Depending on the application, this alone could justify that the models are performing similarly enough to deem them covalid. Consider also that the "difference" model R^2 is only 0.319, which implies that the "difference" regression can only account for about 32 percent of the variation of the responses about the mean. The apparent lack-of-fit suggested by this low R^2 could be used as further evidence of the covalidity of the models.

The important consideration here is that each application of this method is different and that each requires its own specific qualifications for covalidity. No one statistic can provide a blanket "goodness-of-covalidity" measure, and it is my contention that attempting to develop such a statistic would undermine the uniqueness of each specific application of this method. Comparing two (or more) models should occur across many applicable fronts, if possible. Further, the degree of closeness required for a declaration of favorable comparison will differ from case to case, as well. Actually requiring a "difference" model to be zero in order to declare two models covalid should be reserved only for applications in which it is required to have identical models. Some lesser degree of closeness should be sufficient for most applications. This level of closeness may be described by the significance of the "difference" model coefficients, by the R^2 statistic, or by some other objective statistic relevant to the particular case.

IV. Results

Introduction

This chapter documents the performance of the covalidation methodology on the actual MASS and NRMO models at two distinct scenarios. I developed a scenario and translated it into two sets of inputs, one for each model. Further, a second scenario (as well as applicable input sets) was created from the first in order to provide an indication of covalidation at more than a single input point. Each scenario was exercised using the method of output/input crossflow until a convergence of input values was reached. This iterative method was conducted three times for each scenario using different feedback functions in each case. Finally, each scenario was used as the center of a 2^6 plus center point experimental design for gradient comparison.

This chapter is organized much as the basic modeling paradigm presented in Figure 1. The first section corresponds to block 1) in Figure 1; the basic scenario developed by modelers at AMCSAF is summarized. The following section, representing block 2) in Figure 1, presents the setup of the output/input crossflow with respect to two distinct feedback functions. The next section covers the experimental design used to conduct the gradient analysis applied to the result from the output/input crossflow application (block 3 in Figure 1). The results from the actual performance of the output/input crossflow method are presented in the subsequent section, followed by a section concerning the actual performance of the gradient analysis, and finally, the decision as to covalidation is discussed.

The Scenarios

The notional scenario used for this research uses a fleet of 160 military aircraft mixed among C-5s, C-17s, and C-141s, plus several wide body passenger (wbp) aircraft from the Civil Reserve Air Fleet (CRAF). These aircraft are used to fly some 26,000 short tons (stons) of cargo and 35,700 passengers from five onload bases (primarily McGuire and Charleston Air Force Bases (AFB) for the military aircraft and John F. Kennedy (JFK) International Airport for the CRAF) to six offload bases (mainly Bahrain, Dhahran, and King Abdul Aziz International Airports in Saudi Arabia) over a 20-day period. Each cargo mission from the continental United States (CONUS) is flown either direct to its destination, or through one of four en route bases in Europe, specifically Mildenhall, England; Ramstein, Germany; and Moron and Rota, Spain (the passenger carrying CRAF aircraft use separate civil airports in Europe as their en route stops).

The scenario contains delivery requirements through the 20th day after the start of the scenario, but I only collect data for the first 15 days. This is done so that I may obtain statistics from the initial surge of cargo requirements into the theater, while not implying that cargo requirements end after 15 days. The cargo requirements are mixed among bulk (palletized) cargo, over-sized cargo (cargo that will not fit on a pallet, but will fit on a C-130, C-141, or larger aircraft), and out-sized cargo (cargo that will not fit on a C-130 or C-141 aircraft, but will fit on a C-5 or C-17 aircraft).

The initial aircraft fleet is a combination of 60 C-5 aircraft, 50 C-17 aircraft, 50 C-141 aircraft, and 25 wide body passenger CRAF aircraft. These aircraft are first available for missions on the first two days of the scenario. In MASS, the initial location of the military aircraft is specified as either McGuire or Charleston AFB, and the 25 CRAF aircraft begin at JFK International Airport. In NRMO, the initial location of aircraft is not specified, and the model operates such that the aircraft are initially located where they are first required.

The infrastructure available at the various bases used in the MASS and NRMO models are for the most part implicitly assumed in the values of the input variable "MOG." This MOG value (often referred to as "working MOG") is intended to provide a measure of the maximum number of aircraft a base can support in terms of runway usage, taxiing, parking, loading, unloading, and refueling during a designated period of time. One important base function that is explicitly modeled is the daily total fuel available at the base (as opposed to the number of fuel trucks or fuel pumps available, which is included in MOG). Depending on assumptions of the availability of resources, each base is given either a particular value for its MOG and fuel, or a value that indicates the resource is unconstrained. Infrastructure at bases in the United States is considered unconstrained. This assumption implies that both MOG and fuel at CONUS bases are unlimited. Bases in Saudi Arabia are considered to have unlimited fuel, but limited MOG. The en route bases in Europe have both limited fuel and MOG.

The modified scenario, used to provide a look at covalidation at a second point, was created using the base scenario as a starting point. The delivery requirements were then increased by 50 percent to create a total requirement of 39,000 short tons of cargo and 53,500 passengers delivered. Other than the delivery requirements, the scenarios are identical. However, this does not imply that the iterative method converges to the same fixed point. For reference, the scenarios are referred to as the "small scenario" and the "big scenario." See Appendix B to view the pertinent inputs used by the MASS and NRMO models for the small and big scenarios.

The Iterative Method of Output/Input Crossflow

The performance of the iterative method on two or more models is highly dependent upon the choice of feedback functions used in the iterations. With guidance from AMCSAF, I examined the input of each model to determine parameters (in NRMO) whose values had little justification and variables (in MASS) that could benefit from the optimizing nature of a linear program (LP). In addition to determining input parameters and variables that could be improved upon, I needed to find parameters and variables that could be derived from the opposing model's output. The crossflows between MASS and NRMO arrived at for this study include the mix of military aircraft type which may be derived from NRMO and input to MASS, and the MOG efficiency parameter in NRMO which may be derived from MASS output.

Mix of Military Aircraft. The initial fleet of military aircraft utilized in the scenario consisted of 60 C-5s, 50 C-17s, and 50 C-141s. MASS utilizes these allocated aircraft as they become available, without determining whether a particular aircraft is the best choice to transport a particular cargo requirement or not. All aircraft are, therefore, used in approximately the same proportions as their availability and utilization (ute) rates allow. (The ute rate dictates a percentage of time flying which a particular aircraft type may not exceed when averaged over a model specified period of time.)

This method of scheduling aircraft for missions has proven historically not to be optimal. For instance, AMCSAF conducted a study where the addition of an aircraft type (without the removal of any other aircraft) actually reduced throughput during a scenario, presumably because the new aircraft type was not as efficient as the aircraft already included in the study.

It is considered reasonable, then, that altering the mix of aircraft available to a MASS study might similarly alter the throughput realized by the study. An optimal aircraft mix is not necessarily what the NRMO model provides, however. NRMO starts its run with an initial mix of aircraft, just as MASS. However, NRMO is free to use (or not use) each aircraft in any way it deems appropriate for the movement of cargo. It may be implied that this method of aircraft use suggests a more efficient aircraft mix than is found by using the aircraft in the proportions in which they are provided. Therefore, I determine the percentage of all military missions that each military aircraft type fly as output from NRMO. Then, assuming the total size of the military fleet used (160 aircraft) remains constant, I determine a new number of each type of military aircraft to be used by the MASS model using Equation (26). For i = {C-5, C-17, C-141},

$$\mathrm{Planes}_i^{new} = \min\left(\mathrm{round}\left(\frac{\mathrm{Msn}_i}{\mathrm{TotalMsn}} \times 160\right), \mathrm{Avail}_i\right) \tag{26}$$

In Equation (26), Planes_i^{new} is the number of aircraft type *i* to be used as MASS input during the next iteration; Msn_i is the number of missions performed by aircraft type *i* during the first 15 days of the scenario (from NRMO); TotalMsn is the total number of missions performed by all military aircraft types during the first 15 days of the scenario (from NRMO); and Avail_i is the maximum number of aircraft type *i* available for use.

The min function simply takes the least of the two arguments, but the round function used is somewhat different than a typical round function. Since there are three aircraft types and the sum of the Planes_i^{new} must be 160, the round function rounds the two aircraft types closest to an integer to that integer, while the remaining Planes_i^{new} is evaluated such that the sum of all Planes_i^{new} equals 160.

Note that if one of the military aircraft types, C-17 for instance, returns $\text{Planes}_{\text{C-17}}^{new}$ = $\text{Avail}_{\text{C-17}}$, Equation (26) is not sufficient to evaluate the other aircraft types since the total number of aircraft must remain constant at 160. In this case, the other aircraft types must use Equation (27) to evaluate the new number of that aircraft type to be employed in the next iteration. I leave open the possibility of more than one aircraft type equaling its maximum number available, though this may not be possible due to the value of the Avail_i constants. For $k = \{k \in i \ni \text{Planes}_k^{new} = \text{Avail}_k\}, j = i - k$,

$$\text{Planes}_j^{new} = \text{round}\left(\frac{\text{Msn}_j}{\text{TotalMsn} - \sum_k \text{Msn}_k} \times \left(160 - \sum_k \text{Planes}_k^{new}\right)\right) \tag{27}$$

The necessary assumption concerning the input set is that it be compact. The assumption concerning the feedback function is that it is continuous. Any input that does not change due to a feedback function is constant, and therefore is compact. Also, the set of potential aircraft mixes must be compact. The aircraft mix consists of three inputs: the number of C-5s, the number of C-17s, and the number of C-141s. Each of these is an integer value with a minimum of 0 and a maximum of the lesser of the number of that type of aircraft available and 160 (the total size of the allowable fleet), so the set of variables representing the numbers of military aircraft is compact.

It is not as easy to determine the validity of the assumption that mandates continuity of the function that transforms the NRMO output of aircraft mix to MASS. The actual output from NRMO consists of the number of missions flown by day, by aircraft (which may not, in NRMO, be integer valued). The percentage of missions each type of military aircraft flew as compared to all military missions, then, is known and continuous, with a minimum of 0 and a maximum of 1. These percentages multiplied by the total number of aircraft in the fleet (up to a maximum of the number of that aircraft type available for use by the scenario) should provide the number of each military aircraft type I wish to use as the initial fleet mix in MASS. However, since MASS will not accept fractional aircraft as input, an integer must be assigned to each value. This presents a problem since the feedback function is no longer continuous. However, we may contrive the "rounding" function to look continuous, where, in a one-dimensional example, the "steps" of the function which maps real numbers to their closest integers are actually not steps and have derivatives at the step points that are less than infinity. The output of NRMO being continuous, this should not pose a problem since the probability of landing exactly on one of the steps is zero.

MOG Efficiency. The function used to feedback MASS output to NRMO input is the MOG efficiency of the three primary offload bases. MOG efficiency in NRMO is a measure of how efficiently we anticipate a particular base is able to utilize its MOG. This number is an attempt to compensate for the fact that parking spaces, fuel trucks and stations, MHE, taxiways, etc. cannot be optimally scheduled and utilized in real-life operations. Currently, NRMO runs are devised so that this MOG efficiency value is 0.68 for all bases (though the model is capable of separate MOG efficiency values for each base). The value 0.68 is the multiplication of two separate MOG efficiencies: 1) 0.8 which accounts for scheduling inefficiencies and 2) 0.85 which accounts for aircraft queuing inefficiencies. These values may or may not provide a reasonable estimate of the inefficiencies of working MOG on average, but they almost certainly do not accurately reflect the inefficiencies experienced by every base, as these inefficiencies are likely to differ from base to base.

Deriving the MOG efficiency feedback function from MASS output to NRMO input is not a straightforward task, however, and the actual function is open to some debate. In fact, two forms for the feedback function were used for this research. Some of the MASS outputs available which could be considered useful in the creation of such a feedback function include the minimum, average, and maximum numbers of aircraft which are present at each base and the numbers of daily landings and takeoffs that occur at each base. Also of use is the input of the amount of MOG available at each base.

The first MOG efficiency feedback function used in this research is the average number of planes at a base divided by the actual amount of MOG available at a base. For a base that is considered a bottleneck, this function should provide an indication of the proportion of planes that can actually be serviced at a base to the theoretical maximum number of planes capable of being serviced at the base, or the efficiency of MOG at the base. In MASS (and NRMO), MOG is divided into wide-body (C-5) and narrow-body (C-17 and C-141) MOG values that compete with each other for the total MOG at a base.

In order to combine these values into one value that could be used to compare against the average number of aircraft present at the base, I adjusted the theoretical MOG at each base by the proportion of narrow-body to wide-body aircraft that actually landed at the base. For instance, at a base with a MOG of 26 narrow-body aircraft or 13 wide-body aircraft (as was used for the MOG at Bahrain in this research), I multiplied the proportion of narrow-body aircraft landings (to all landings) by 26 and added this to the proportion of wide-body aircraft landings multiplied by 13. This method of determining an equivalent MOG value for this example base is illustrated in Figure 12.

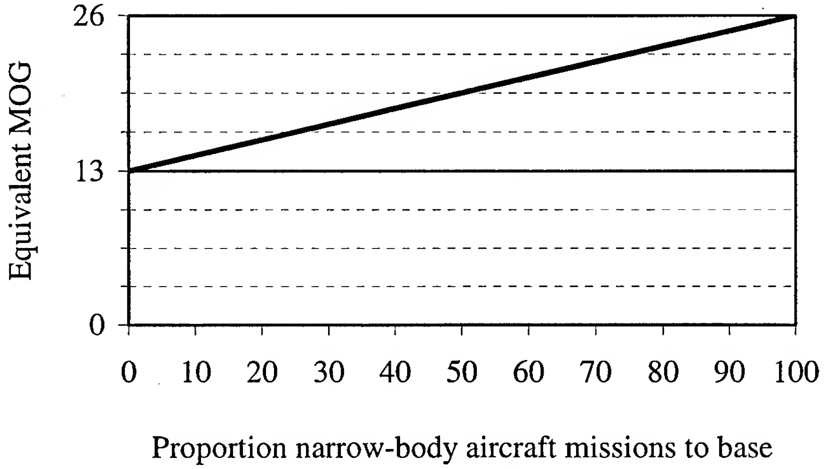

Figure 12: Equivalent MOG at a Typical Base

The result of this feedback function is the theoretical maximum number of aircraft on the ground at the base given that aircraft arrive to the base in the determined proportion of narrow to wide bodies. This feedback function is shown as Equation (28). For j = {OBBI, OEDR, OEJN} (the primary offload bases),

$$\text{MOGEffavgact}_j^{new} = \frac{\text{Avgplanes}_j}{\frac{\text{NBlandings}_j}{\text{ALLlandings}_j}\times\text{NBMOG}_j + \frac{\text{WBlandings}_j}{\text{ALLlandings}_j}\times\text{WBMOG}_j} \tag{28}$$

In Equation (28), $\text{MOGEffavgact}_j^{new}$ is the MOG efficiency of base *j* to be used as NRMO input during the next iteration ("avgact" denotes that I take the *average* number of planes divided by the *actual* MOG at the base, and the term may be used to refer to this feedback function); Avgplanes$_j$ is the average number of planes at base *j* over the course of the scenario (from MASS); NBlandings$_j$ (WBlandings$_j$) is the number of narrow-body (wide-body) aircraft landings at base *j* during the scenario (from MASS); ALLlandings$_j$ is the total of narrow-body and wide-body aircraft landings at base *j* during the scenario (from MASS); and NBMOG$_j$ (WBMOG$_j$) is the amount of MOG available for use at base *j* by narrow-body (wide-body) aircraft.

Again, an assumption for the input set is compactness. The inputs to NRMO that do not vary due to feedback from NRMO are constant and therefore compact. The feedback of MOG efficiency requires that the set of possible MOG efficiency values be compact. Of course, the minimum possible MOG efficiency value is 0.0. The upper bound could be 1.0 if the MOG value actually represented a hard constraint. However, it is possible that more than the MOG amount of aircraft may use a base at a given time in the MASS model, so the upper limit of MOG efficiency may be greater than 1.0. However, an arbitrary upper bound (which at the very most could be represented by the total number of aircraft in the fleet divided by the allowable MOG at a base) could be enforced if required.

The assumption of continuity of the feedback function causes no problems in this case, since we divide a real number by a real number and use the resulting real number directly as the output of the feedback function.

The second function used to feedback MOG efficiency from MASS to NRMO is similar to the first, but uses MASS output to necessitate MOG efficiency values between 0.0 and 1.0 (inclusive). In this case, the average number of aircraft on the ground is divided by the average (over the days of the study) of the maximum number of aircraft on the ground (for each day). Of course, the average for any given day cannot be greater than the maximum for that day, so the averaging of each over the length of the simulation cannot produce a MOG efficiency of greater than 1.0. Since MASS may, at times, place more aircraft on the ground at a base than is feasible, it is considered that this MOG efficiency function (given in the following equation) may better indicate the actual MOG efficiency for a base.

$$\text{MOGEffavgmax}_j^{new} = \frac{\text{Avgplanes}_j}{\sum_{d=0}^{days} \text{Maxplanes}_{j,d} \Big/ days} \qquad (29)$$

In Equation (29), $\text{MOGEffavgmax}_j^{new}$ is the MOG efficiency of base j to be used as NRMO input during the next iteration ("avgmax" denotes that I take the *average* number of aircraft divided by the average of the daily *maximum* number of aircraft at the base, and the term may be used to refer to this feedback function); Avgplanes$_j$ is the average number of planes at base j over the course of the scenario (from MASS); *days* is the number of days considered in the scenario; and Maxplanes$_{j,d}$ is the maximum number of aircraft on the ground at base j on day d (from MASS).

The assumptions of compactness and continuity in the feedback function cause no problems in this case. The MOG efficiency is limited to real values between 0.0 and 1.0 (inclusive), and the feedback function in this case is clearly continuous (for the same reasons as the other case of MOG efficiency feedback).

The output/input crossflow method for my application is depicted as in Figure 13.

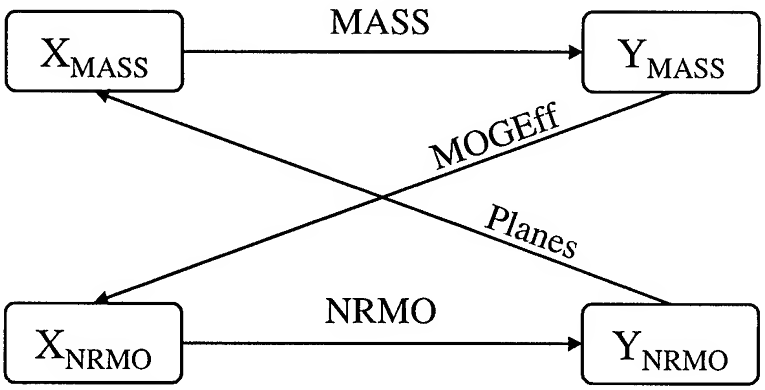

Figure 13: Relationship between MASS and NRMO

Experimental Design

Recall that we wish to construct metamodels across a small experimental design of relevant input variables to provide a convenient means of effecting a rational comparison of the models. I constructed two experimental designs; one around the input point of convergence corresponding to the small scenario and one around the input point of convergence of the big scenario. I employed system experts at AMCSAF to assist in the development of the pertinent input variables used in the experimental design. An important factor to realize is that each input variable we wish to use as a factor in the experimental design must be present as an input variable into each model.

First, the number of military aircraft available for use was varied. Regardless of the breakdown of C-17, C-5, and C-141 aircraft arrived at during the iterative procedure, the total size of the military fleet is 160 aircraft. I varied the size of this total fleet by as much as ten percent by varying the number of C-5 and C-17 aircraft by eight aircraft each from the base case (as described later, the iterative method drove the number of C-141 aircraft to zero, so I did not vary this input variable from that value). In this way, if the number of both C-5s and C-17s are high (low), the total fleet is ten percent larger (smaller) than the base case. If the number of C-5s is high while the number of C-17s is low, or vice-versa, the total number of aircraft in the fleet is the same as that in the base case, regardless of the specific number of C-5s or C-17s in the fleet.

The other variables I used as factors in the experimental design are the fuel available to military aircraft at the four en route bases. The base amount of fuel at these en route bases is 800,000 gallons per day available at each Mildenhall, Ramstein, and Moron, and 250,000 gallons per day available at Rota. The total amount of fuel available per day at the en route bases, then, is 2,650,000 gallons. Using a method similar to the varying of aircraft, I wish to vary the amount of total fuel available by some small relative amount; I selected a change representing just under ten percent of the total fuel available; 249,000 gallons per day. Spread over the four en route bases, a "low" value decreases the amount of fuel at a base by 62,250 gallons, and a "high" value increases the fuel at a base by 62,250 gallons. Even though the percent change from the base value to a "high" or "low" value is different for Rota than for the other bases, if all bases are "high," the overall increase in fuel is almost ten (~9.4) percent, and if two bases are "high" while two bases are "low," the total amount of fuel at the en route bases is equal to the amount of fuel available in the base case. Similarly, if three bases' fuels are "high" while the other is "low," the total daily fuel available at the en route bases is almost five (~4.7) percent greater than the base case.

With six factors to vary, a full-factorial design yields a total requirement of 2^6, or 64, design points. Including the center point in the design, each experimental design (the small and big scenarios) consists of 65 design points, which are run on the MASS and NRMO models for comparison. I run 30 replications of the MASS model at each design point to narrow the confidence region established by the created metamodel. Table 4 below shows the coded design using –1 for "low" values and +1 for "high" values. In the table, the en route bases are identified by their International Civil Aviation Organization (ICAO) identifiers where EDAR refers to Ramstein, Germany; EGUN is Mildenhall, England; LEMO is Moron, Spain; and LERT is Rota, Spain.

Table 4: Experimental Design for MASS and NRMO Models

i	Number of Aircraft		Fuel at En Route Bases			
	C-5	C-17	EDAR	EGUN	LEMO	LERT
1	-1	-1	-1	-1	-1	-1
2	+1					
3	-1	+1				
4	+1					
5	-1	-1	+1			
6	+1					
7	-1	+1				
8	+1					
9	-1	-1	-1	+1		
10	+1					
11	-1	+1				
12	+1					
13	-1	-1	+1			
14	+1					
15	-1	+1				
16	+1					
17	-1	-1	-1	-1	+1	
18	+1					
19	-1	+1				
20	+1					
21	-1	-1	+1			
22	+1					
23	-1	+1				
24	+1					
25	-1	-1	-1	+1		
26	+1					
27	-1	+1				
28	+1					
29	-1	-1	+1			
30	+1					
31	-1	+1				
32	+1					
33-64	Repeat above experiments with LERT = +1					+1
65	0	0	0	0	0	0

Performance Results— Output/Input Crossflow Method

This section examines the actual performance of the output/input crossflow method on the MASS and NRMO models with respect to the small and big scenarios. Recall that different forms of the feedback functions are also considered. The first three subsections examine various feedback functions on the small scenario, and the next three subsections cover the same feedback functions on the big scenario.

Military Aircraft Mix and "avgact" MOG Efficiency Feedback Functions. The first application of the output/input crossflow method on the small scenario uses the military aircraft fleet mix feedback function, Equations (24) and (25), from NRMO to MASS and the "avgact" MOG efficiency feedback function, Equation (28), from MASS to NRMO. Recall that the MOGEffavgact function implies that the average number of aircraft at a base is divided by the actual available MOG at the base to determine the MOG efficiency at the base, allowing for the possibility of a MOG efficiency greater than 1.0. Also, I did not specify a particular value for the Avail (maximum number of aircraft of a particular type available, see Equation (26)) variables in this application of the output/input crossflow method, preferring to allow the number of planes to be modified to the extent that the method dictates (of course, this has the effect of setting Avail for each aircraft type to 160, or the number of aircraft in the entire fleet).

Table 5 displays how the inputs changed through the iterations. In the table, the inputs to NRMO are again labeled with ICAO identifiers, where OBBI represents Bahrain, OEDR is Dhahran, and OEJN refers to King Abdul Aziz. The initial inputs are displayed as iteration 1. From the first couple iterations, it can be seen that the new values for MOG efficiency are influenced by the specific fleet mix used in the MASS model, while the fleet mix is not largely affected by the MOG efficiency values used in NRMO. The input for the second iteration shows that both models' inputs are significantly changed due to the performance of the other model. This initial change, however, is due to the entire input set used by the other model. The MASS input for the third iteration is only slightly modified, even though NRMO's MOG efficiency had been significantly altered in the second iteration, while the NRMO input for the third iteration changes drastically due to the significant change in the MASS fleet mix from iteration two. Upon reaching these iteration three values, the changes made to the input values decreases to near insignificance, and full convergence of the values occurs by iteration nine.

Table 5: Feedback Values: avgact

To:	MASS			NRMO		
Feedback:	Planes			MOGEffavgact		
Iteration	C-17	C-5	C-141	OBBI	OEDR	OEJN
1	50	60	50	0.680	0.680	0.680
2	35	108	17	0.680	1.104	1.140
3	34	108	18	1.503	0.424	0.812
4	34	106	20	1.496	0.446	0.817
5	33	106	21	1.497	0.492	0.762
6	34	106	20	1.518	0.515	0.785
7	34	107	19	1.497	0.492	0.762
8	34	106	20	1.478	0.473	0.779
9	34	106	20	1.497	0.492	0.762
10	34	106	20	1.497	0.492	0.762

The NRMO model proves to be relatively insensitive to changes in the MOG efficiency factor, since, though I end up with MOG efficiency values greater than 1.0, and though these values shifted significantly, I see very little suggested change to the

MASS input based on these facts (or in the output of interest, throughput). Figure 14 shows the convergence of each models' inputs graphically. In Figure 14 (and in all similar figures that follow), the values from the first iteration are plotted on the left side of the figure. The final two iterations in each figure display the same data and demonstrate the convergence. Also, in order to include all the values on the same scale, I show the proportion of the fleet for each of the aircraft types, instead of the raw value.

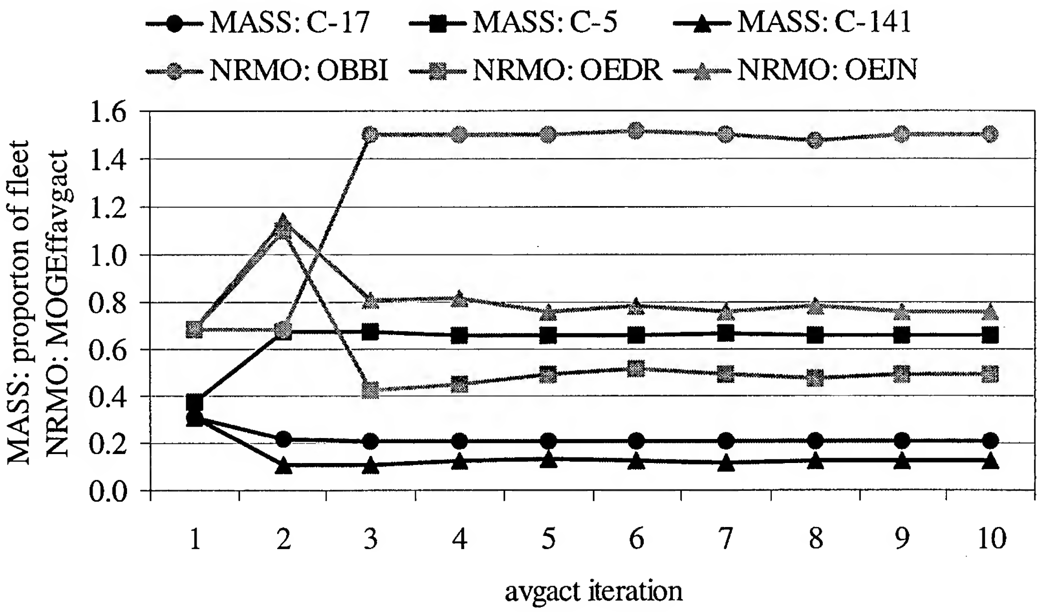

Figure 14: Input Convergence: avgact

Table 6 displays a comparison of the main output metric of concern, total short tons (stons) delivered in the first 15 days of the scenario. Also shown are the percentage differences between MASS and NRMO for the three metrics that comprise total stons: bulk stons, oversize stons, and outsize stons. Of course, the outputs for each of the models converge as the inputs do, a fact that does not imply that the outputs of the two models converge to the same value (which, as seen from the table, they do not). The difference between MASS and NRMO in terms of total stons, in fact, hovers around 25.5 to 27.5 percent, and I notice that neither the MASS nor the NRMO output varies much throughout the iterations. Further, MASS is delivering more cargo than NRMO, which seems counter-intuitive since one might expect the optimization model to deliver more efficiently.

Table 6: Output Convergence: avgact

Iter.	Total stons/day		MASS is ___% greater than NRMO			
	MASS	NRMO	Bulk	Over	Out	Total
1	1632.2	1280.0	28.0%	30.3%	20.0%	27.5%
2	1625.6	1281.5	26.3%	30.5%	19.0%	26.9%
3	1624.9	1294.4	25.3%	28.3%	19.2%	25.5%
4	1623.1	1293.8	23.4%	29.6%	18.9%	25.5%
5	1624.2	1291.2	26.3%	27.7%	20.2%	25.8%
6	1623.1	1286.6	27.4%	27.7%	20.2%	26.2%
7	1625.6	1279.2	29.8%	27.8%	20.7%	27.1%
8	1623.1	1289.6	26.0%	28.6%	18.9%	25.9%
9	1623.1	1291.2	26.2%	27.6%	20.2%	25.7%
10	1623.1	1291.2	26.2%	27.6%	20.2%	25.7%

These discrepancies, the lack of output variation between successive iterates, the large output difference between MASS and NRMO, and the fact that the simulation is delivering more cargo than the optimization model, can all be justified with one explanation: The scenario's movement requirements are not demanding to either model. Neither model has a problem with handling such a scenario; however, each model handles the situation differently. Recall that MASS delivers requirements using the first available aircraft searching for the first available cargo requirement and carrying it using the first available route. Therefore, if the model has delivered all required cargo by the appropriate date, the model will still search for the next available piece of cargo and deliver it, effectively getting ahead on its required deliveries.

Conversely, NRMO operates in an attempt to minimize non-deliveries and late deliveries of cargo and maximize a small bonus for aircraft remaining at their home-station. As a result, if NRMO is "caught up" with its required deliveries, the action it takes to improve its objective function value is not to go after more cargo, but rather to let aircraft sit idle at their home station. For this reason, NRMO only attempts to move cargo when it would benefit the objective function to do so. Figure 15 graphically depicts the lack of variation in output from one iteration to the next for the three size-specific cargo throughput metrics. The indication in this stable output is that neither model has any problem in meeting the requirements of the scenario TPFDD. The figure also reveals that MASS throughput exceeds NRMO throughput consistently across cargo types.

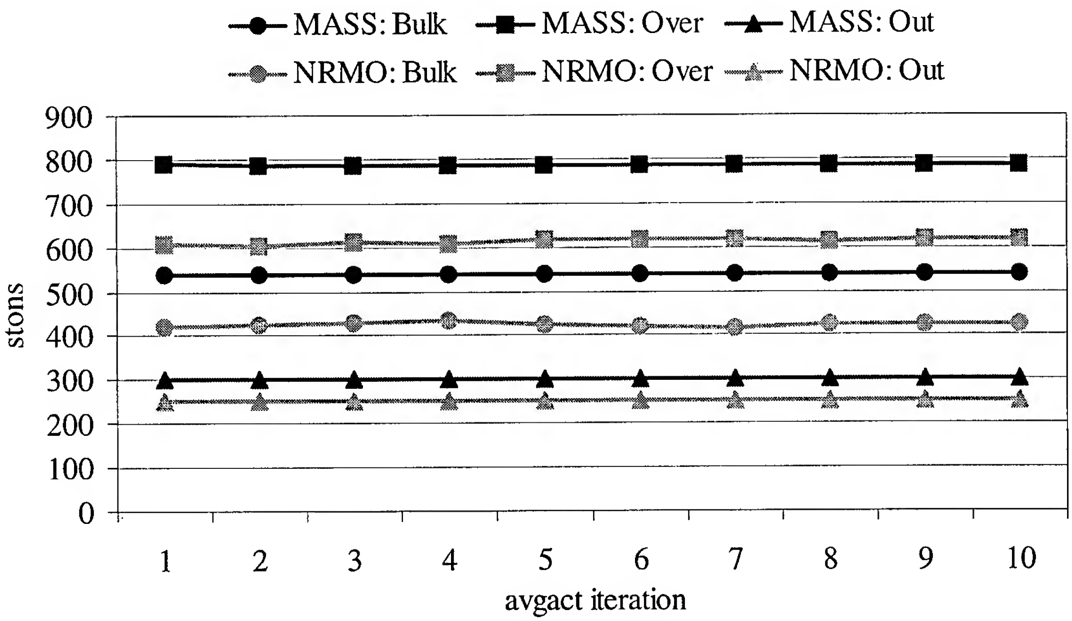

Figure 15: Output Convergence: avgact

Military Aircraft Mix and "avgmax" MOG Efficiency Feedback Functions. The next application of the output/input crossflow method on the small scenario uses the military aircraft fleet mix feedback function, Equations (24) and (25), from NRMO to MASS and the "avgmax" MOG efficiency feedback function, Equation (29), from MASS to NRMO. Recall that the MOGEffavgmax function implies that the average number of aircraft at a base is divided by the maximum number of aircraft at the base to determine the MOG efficiency at the base, forcing MOG efficiency values to be less than or equal to 1.0. Table 7 displays the changing inputs, while Figure 16 shows the convergence in six iterations graphically.

Table 7: Feedback Values: avgmax

To:	MASS			NRMO		
Feedback:	Planes			MOGEffavgmax		
Iteration	C-17	C-5	C-141	OBBI	OEDR	OEJN
1	50	60	50	0.680	0.680	0.680
2	35	108	17	0.635	0.594	0.645
3	35	108	17	0.743	0.420	0.565
4	34	106	20	0.743	0.420	0.565
5	34	106	20	0.737	0.459	0.578
6	34	106	20	0.737	0.459	0.578

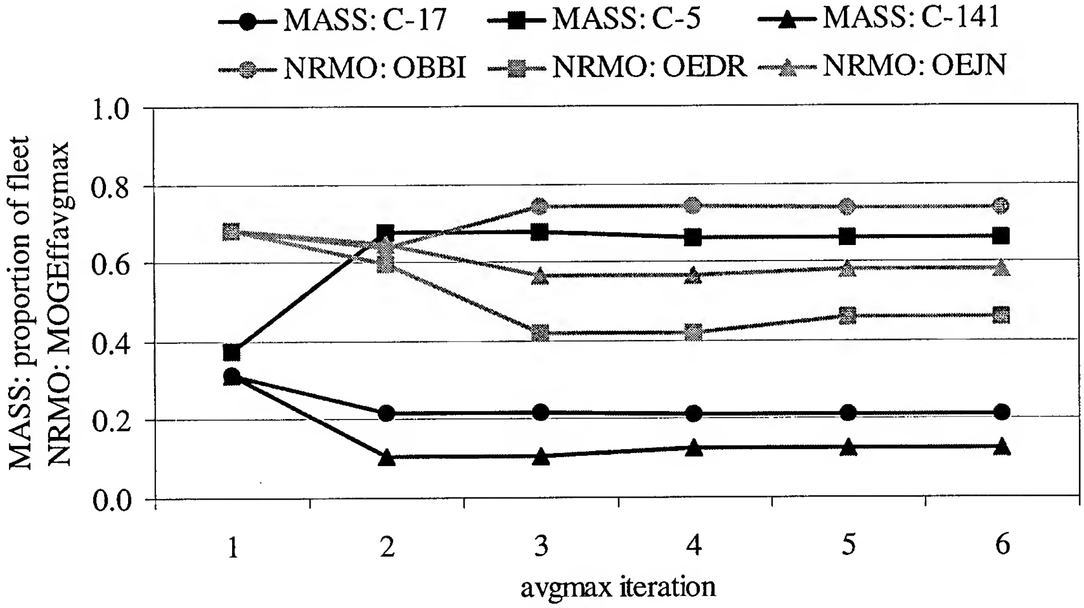

Figure 16: Input Convergence: avgmax

Table 8 again shows the relative lack of change in the response of total throughput across the iterations while Figure 17 displays this same phenomena applied to the three size specific output metrics graphically.

Table 8: Output Convergence: avgmax

Iter.	Total stons/day		MASS is ___% greater than NRMO			
	MASS	NRMO	Bulk	Over	Out	Total
1	1632.2	1280.0	28.0%	30.3%	20.0%	27.5%
2	1625.6	1279.2	29.8%	27.9%	20.4%	27.1%
3	1625.6	1292.9	27.9%	27.0%	19.0%	25.7%
4	1623.6	1292.9	27.8%	26.6%	19.2%	25.6%
5	1623.6	1291.2	26.3%	27.5%	20.6%	25.7%
6	1623.6	1291.2	26.3%	27.5%	20.6%	25.7%

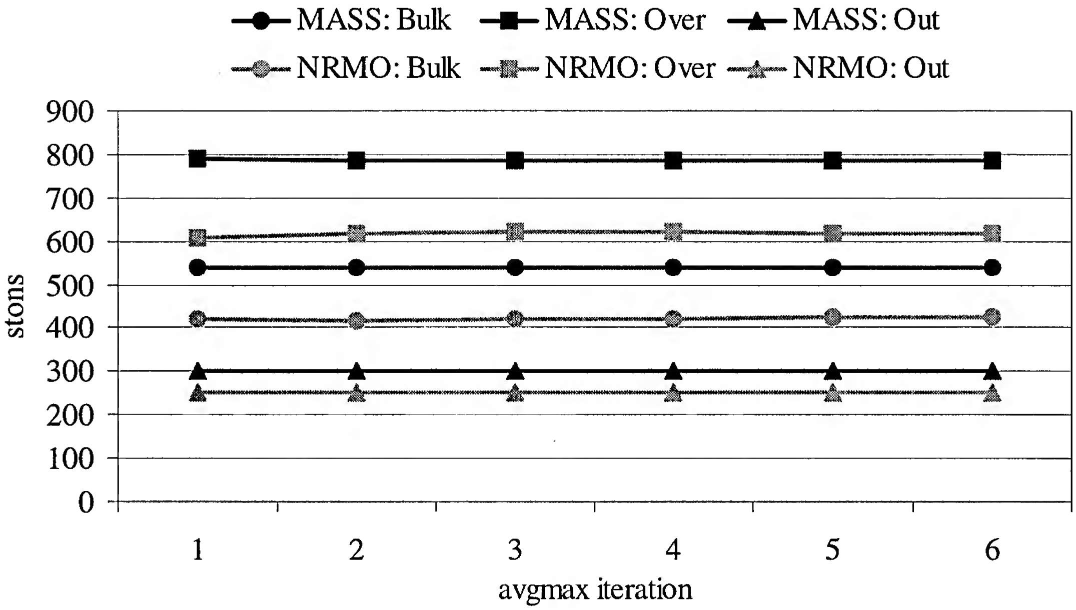

Figure 17: Output Convergence: avgmax

<u>Military Aircraft Mix and "ftsam" MOG Efficiency Feedback Functions</u>. In considering the performance of the experimental design on the MASS and NRMO models, I realized that using either of the preceding sets of convergent inputs would be breaking one of this method's fundamental assumptions that inputs for each model be as analogous as possible. The problem comes in that while I have been modifying the

MASS input with the number of each type of military plane each iteration, I never modify the NRMO input of the number of each type of military aircraft. In the "ftsam" MOG efficiency feedback scheme, I use the "avgmax" feedback function, Equation (29), as the feedback from MASS to NRMO, and I use the same mix of military aircraft feedback function, Equations (24) and (25), as the feedback from NRMO to MASS. Now, however, I also feed the same aircraft numbers back to the NRMO model ("ftsam" refers to *f*eedback-*t*o-*s*elf (regarding the mix of military aircraft feedback function), *a*verage number of aircraft divided by the average of the daily *m*aximum number of aircraft at the base, and the term may be used to refer to this instance). The flow of the output/input crossflow method including self-feedback to the NRMO model is depicted in Figure 18.

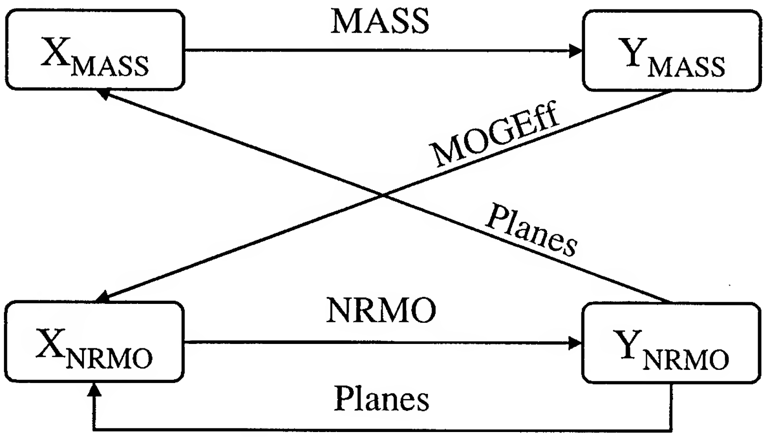

Figure 18: Self-Feedback Relationship between MASS and NRMO

Since this application of the output/input crossflow method is to be used as the center point in the experimental design, I wish to allow only a realistic maximum number of each aircraft type to participate in the movement of cargo. For this reason, I set the Avail (maximum number of aircraft of a particular type available, see Equation (26)) variable from the mix of military aircraft feedback function equal to the maximum number of each type available in the actual USAF inventory. The value for Avail chosen for each aircraft type represents the entire USAF inventory with a few aircraft of each type taken out for training purposes and a few removed to provide back-up aircraft. The final values used for this variable are: $\text{Avail}_{\text{C-17}} = 102$; $\text{Avail}_{\text{C-5}} = 104$; and $\text{Avail}_{\text{C-141}} = 96$.

Table 9 shows the input convergence of the "ftsam" application. In the table, it is shown that the number of C-141s goes to zero. This event is not unexpected since the C-17 and C-5 may carry much more cargo each mission than the C-141. The C-17 hauls over twice the average load of the C-141 (45 stons versus 19 stons) while occupying (to the degree that the models are concerned) the same amount of MOG. The C-5 requires roughly twice the MOG as the C-141 (and C-17), while carrying an average load over three times that of a C-141 (61.3 stons versus 19 stons). Since in NRMO a small bonus is awarded to the objective function for using as few aircraft as possible (i.e., remaining at home station), the most efficient aircraft are used first. In fact, as shown in the table, no C-141s are recommended to move the requirement. Further, C-5 aircraft are preferred to C-17s because, while they use more MOG per short ton of cargo, the requirement is relatively undemanding to move, even while using the added MOG. The number of C-5s, in fact, is driven to the maximum number available ($\text{Avail}_{\text{C-5}}$).

Table 9: Feedback Values: ftsam

To:	MASS/NRMO			NRMO		
Feedback:	Planes			MOGEffavgmax		
Iteration	C-17	C-5	C-141	OBBI	OEDR	OEJN
1	50	60	50	0.680	0.680	0.680
2	37	104	19	0.634	0.577	0.650
3	56	104	0	0.734	0.438	0.571
4	56	104	0	0.725	0.215	0.602
5	56	104	0	0.725	0.215	0.602

Figure 19 displays the convergence of the input values graphically. Notice that with the number of aircraft stabilizing so quickly, the MOG efficiency is forced to stabilize quickly as well, and the convergence occurs in only five iterations. Again we see the robustness of the NRMO model to changes in MOG efficiency.

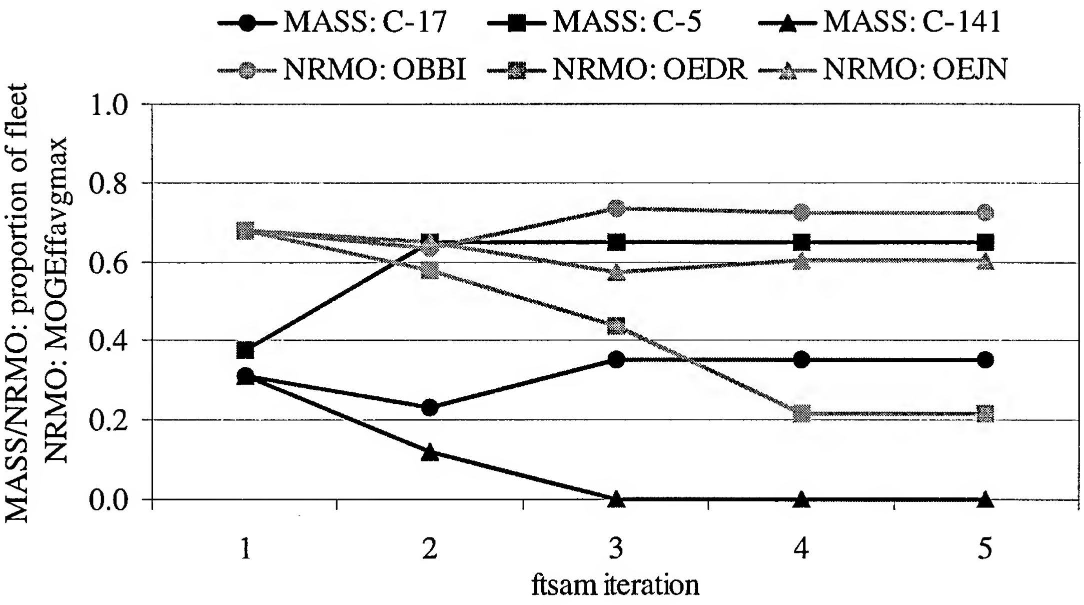

Figure 19: Input Convergence: ftsam

Table 10 displays the output convergence of the "ftsam" application. Of note is the improvement of the models' outputs with respect to each other, ending with total throughput only 15 percent from each other. Figure 20 shows graphically how the oversize and bulk throughput converge while the outsize throughputs diverge slightly.

Table 10: Output Convergence: ftsam

Iter.	Total stons/day		MASS is ___% greater than NRMO			
	MASS	NRMO	Bulk	Over	Out	Total
1	1634.2	1280.0	28.2%	30.4%	20.1%	27.7%
2	1625.9	1338.9	23.1%	18.2%	27.3%	21.4%
3	1616.3	1355.1	23.3%	15.3%	23.3%	19.3%
4	1616.3	1403.7	5.4%	17.6%	28.9%	15.1%
5	1616.3	1403.7	5.4%	17.6%	28.9%	15.1%

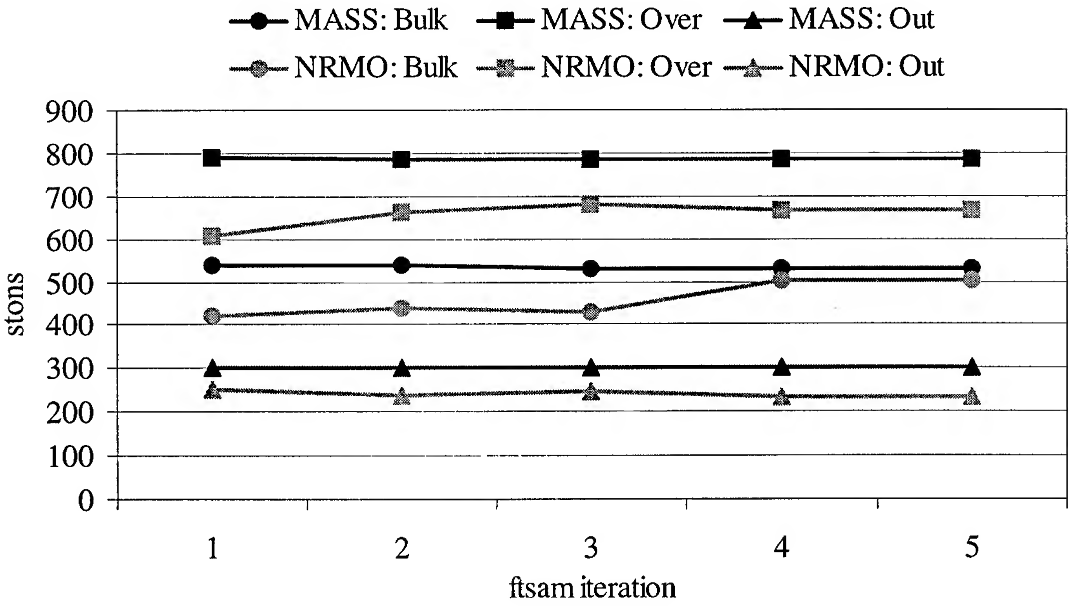

Figure 20: Output Convergence: ftsam

Military Aircraft Mix and “bigavgact” MOG Efficiency Feedback Functions. The following three applications of the output/input crossflow method use identical feedback functions as the first three applications, except they are performed on the big scenario. This big scenario differs from the small scenario only in the size of the requirements in the TPFDD. Each tonnage in the small scenario TPFDD is multiplied by 1.5 to obtain the tonnage in the big scenario.

The first set of feedback functions applied to the big scenario include the mix of military aircraft function, Equations (24) and (25), from NRMO to MASS and the avgact (average number of aircraft at a base divided by the actual MOG available at the base) MOG efficiency function, Equation (28), from MASS to NRMO. The converging input values are shown in Table 11. As observed in the table and in Figure 21, the C-17 is now the preferred aircraft for transporting the cargo requirements. This is a result of the C-17 delivering cargo more efficiently than the C-5 (or C-141) with reference to MOG use. With a higher cargo requirement, the MOG at the bases plays a larger role, and efficiently utilizing that MOG becomes a higher priority for NRMO than in the small scenario. This is the case because in the small scenario, the aircraft can easily carry all of the available cargo, regardless of the MOG; in other words, the amount of available cargo is a constraint. In the big scenario, however, the amount of MOG at the bases becomes a tighter constraint than the amount of available cargo, and NRMO seeks to use the now constrained MOG as efficiently as possible.

Table 11: Feedback Values: bigavgact

To:	MASS			NRMO		
Feedback:	Planes			MOGEffavgact		
Iteration	C-17	C-5	C-141	OBBI	OEDR	OEJN
1	50	60	50	0.680	0.680	0.680
2	76	29	55	0.604	1.121	1.657
3	74	30	56	0.238	0.745	1.714
4	75	30	55	0.250	0.792	1.646
5	75	30	55	0.244	0.770	1.699
6	75	30	55	0.244	0.770	1.699

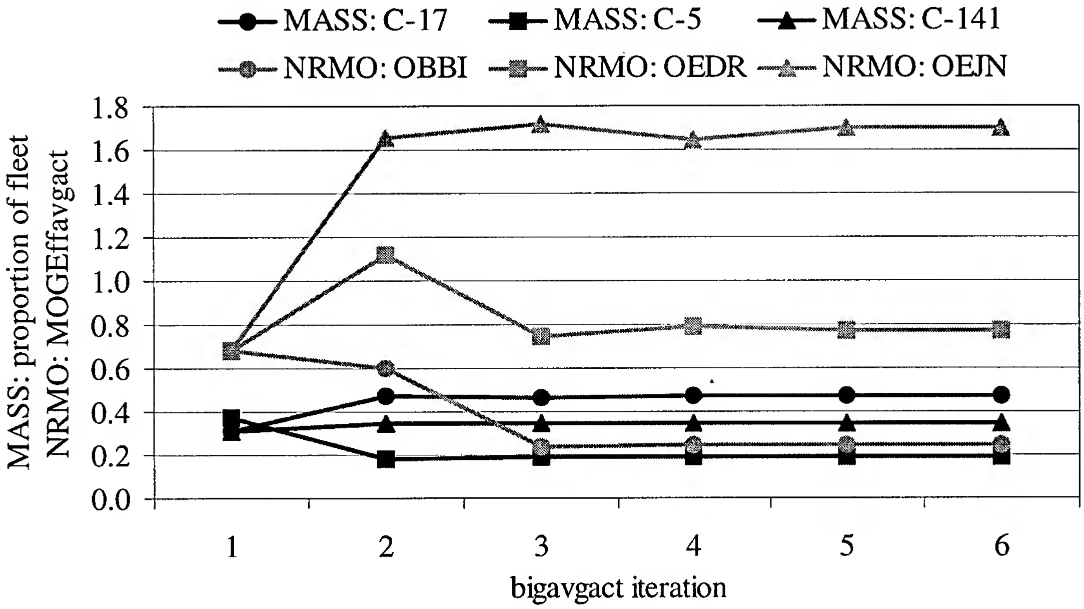

Figure 21: Input Convergence: bigavgact

In Table 12, the throughput realized by the MASS and NRMO models is much closer in the big scenario than it was in the small scenario. The final iteration total throughput difference is only 9.1 percent, as opposed to the 25.7 percent for the small scenario. The reason for this improvement is the same as the motive for creating the big scenario. The big scenario taxes the strategic airlift system more than the small scenario, and the

NRMO model does not have the same opportunity to take advantage of the objective function bonus of not using aircraft (i.e., remaining at home station). So whereas the MASS model continually tries to move the next requirement in the TPFDD, regardless of whether its cargo movement is outpacing this requirement, the NRMO model now finds that it must nearly continually pursue the next piece of cargo in order to keep pace with the requirement. The result is that, in the big scenario, both models are behaving similarly in their pursuit of the next cargo requirement, even though they are acting in that way for different reasons. The fact that NRMO still lags nearly ten percent behind MASS in throughput suggests that the scenario could be even larger and still be handled capably by the models. Figure 22 shows graphically the relative closeness of the throughput of the three cargo types.

Table 12: Output Convergence: bigavgact

Iter.	Total stons/day		MASS is ___% greater than NRMO			
	MASS	NRMO	Bulk	Over	Out	Total
1	2085.4	1879.8	7.5%	5.9%	36.6%	10.9%
2	2018.8	1868.7	5.1%	4.5%	26.2%	8.0%
3	2044.7	1876.1	6.4%	4.4%	30.5%	9.0%
4	2036.3	1871.0	6.0%	4.1%	32.0%	8.8%
5	2036.3	1865.8	6.0%	8.1%	18.8%	9.1%
6	2036.3	1865.8	6.0%	8.1%	18.8%	9.1%

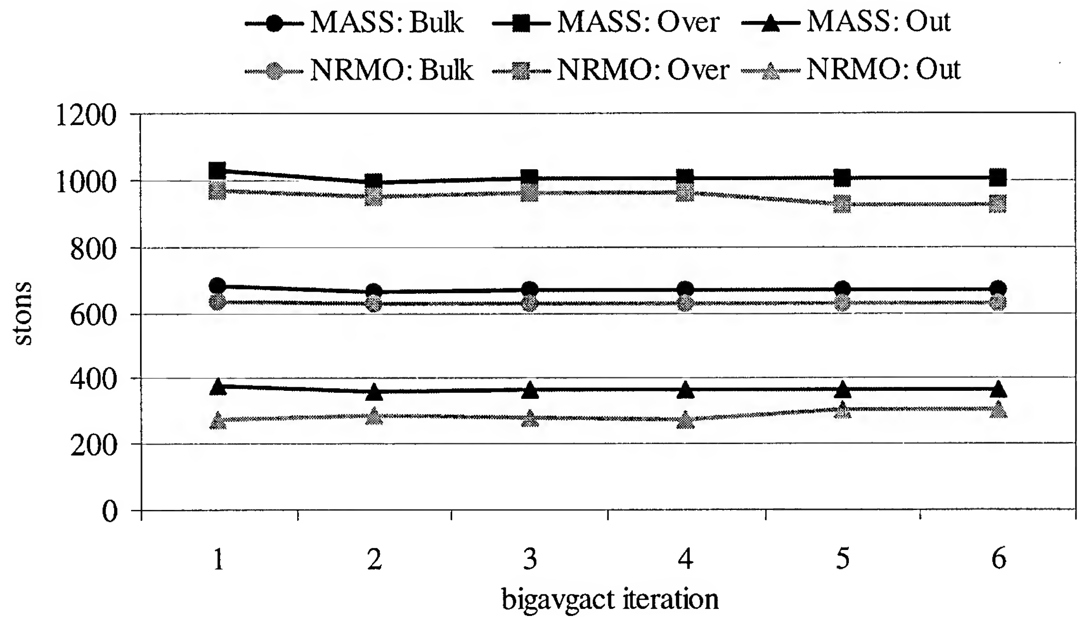

Figure 22: Output Convergence: bigavgact

Military Aircraft Mix and "bigavgmax" MOG Efficiency Feedback Functions. The second application of the big scenario uses the "avgmax" (average number of aircraft at a base divided by the average of the daily maximum number of aircraft at the base) function, Equation (29), to feedback the MOG efficiency value from the MASS model to NRMO. Table 13 and Figure 23 display the results. Once again, the C-17 is preferred over the other two aircraft types due to its efficient use of MOG.

Table 13: Feedback Values: bigavgmax

To:	MASS			NRMO		
Feedback:	Planes			MOGEffavgmax		
Iteration	C-17	C-5	C-141	OBBI	OEDR	OEJN
1	50	60	50	0.680	0.680	0.680
2	76	29	55	0.516	0.568	0.651
3	76	29	55	0.394	0.465	0.597
4	75	30	55	0.394	0.465	0.597
5	75	30	55	0.400	0.464	0.601
6	75	30	55	0.400	0.464	0.601

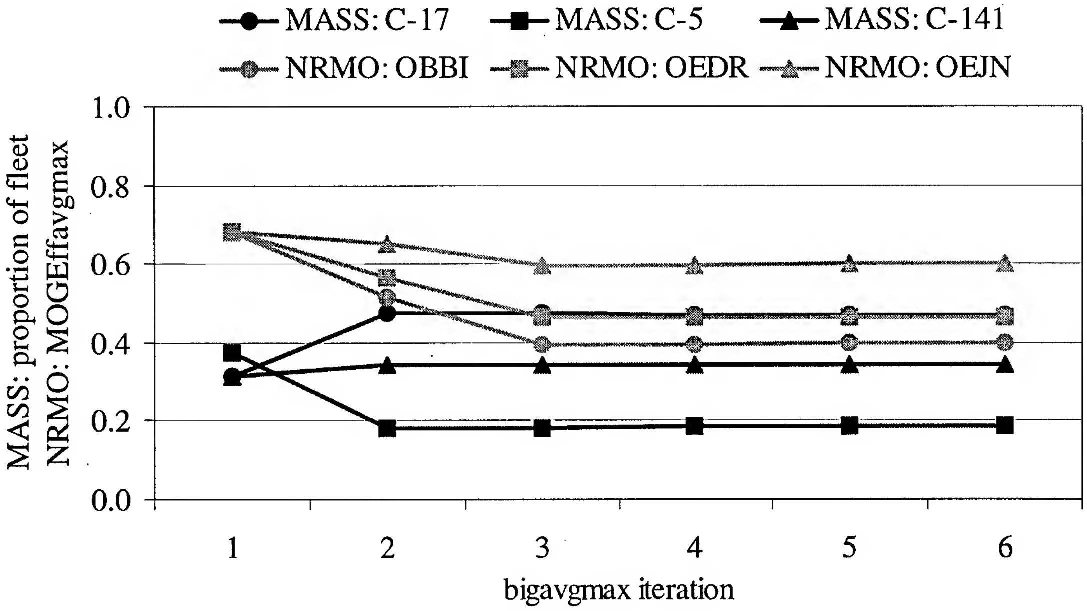

Figure 23: Input Convergence: bigavgmax

The output convergence from this application of the big scenario is displayed in Table 14 and Figure 24. Again the throughput difference is considerably smaller than that for the same feedback functions applied to the small scenario (8.6 percent versus 25.7 percent). The reasons for this are the same as those presented in the previous subsection.

Table 14: Output Convergence: bigavgmax

Iter.	Total stons/day		MASS is ___% greater than NRMO			
	MASS	NRMO	Bulk	Over	Out	Total
1	2085.4	1879.8	7.5%	5.9%	36.6%	10.9%
2	2018.8	1876.8	3.2%	7.3%	17.5%	7.6%
3	2018.8	1874.3	3.1%	7.9%	16.6%	7.7%
4	2036.3	1874.3	4.0%	8.8%	17.9%	8.6%
5	2036.3	1875.2	4.6%	6.4%	24.3%	8.6%
6	2036.3	1875.2	4.6%	6.4%	24.3%	8.6%

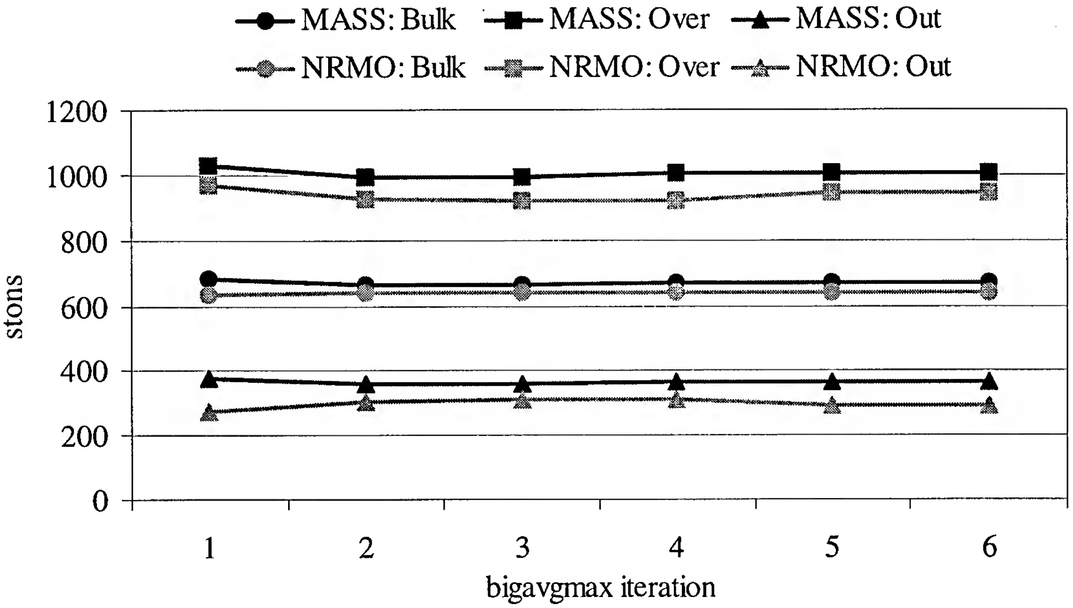

Figure 24: Output Convergence: bigavgmax

Military Aircraft Mix and "bigftsam" MOG Efficiency Feedback Functions. The final application of the output/input crossflow method again uses the "avgmax" MOG efficiency feedback function, Equation (29), from the MASS to the NRMO model, but the NRMO model supplies the result of the mix of military aircraft feedback function, Equations (24) and (25), to both MASS and NRMO. This application (termed "bigftsam" for *big* scenario, *f*eedback *t*o *s*elf, *a*verage number of aircraft divided by the average daily *m*aximum number of aircraft at the base) is performed in order to ensure the inputs are as close as possible for the experimental design, since without the self-feedback of the NRMO mix of military aircraft, the two models would be modeling different fleets of aircraft. Table 15 shows the convergence of the dynamic inputs, while Figure 25 shows the convergence graphically.

Table 15: Feedback Values: bigftsam

To:	MASS/NRMO			NRMO		
Feedback:	Planes			MOGEffavgmax		
Iteration	C-17	C-5	C-141	OBBI	OEDR	OEJN
1	50	60	50	0.680	0.680	0.680
2	76	29	55	0.516	0.568	0.651
3	102	41	17	0.413	0.471	0.599
4	102	53	5	0.464	0.370	0.670
5	102	57	1	0.516	0.286	0.698
6	102	58	0	0.537	0.251	0.710
7	102	58	0	0.533	0.244	0.708
8	102	58	0	0.533	0.244	0.708

The number of C-141s used is again (as in the "ftsam" application) driven to zero. In this application, however, C-17s are preferred to C-5 aircraft. The number of C-17s required is pushed to its maximum ($\mathrm{Avail}_{C\text{-}17} = 102$) because the C-17 delivers cargo more efficiently in terms of MOG use than does the C-5, and the big scenario is demanding enough to require conservation of MOG by the models for the cargo to be moved.

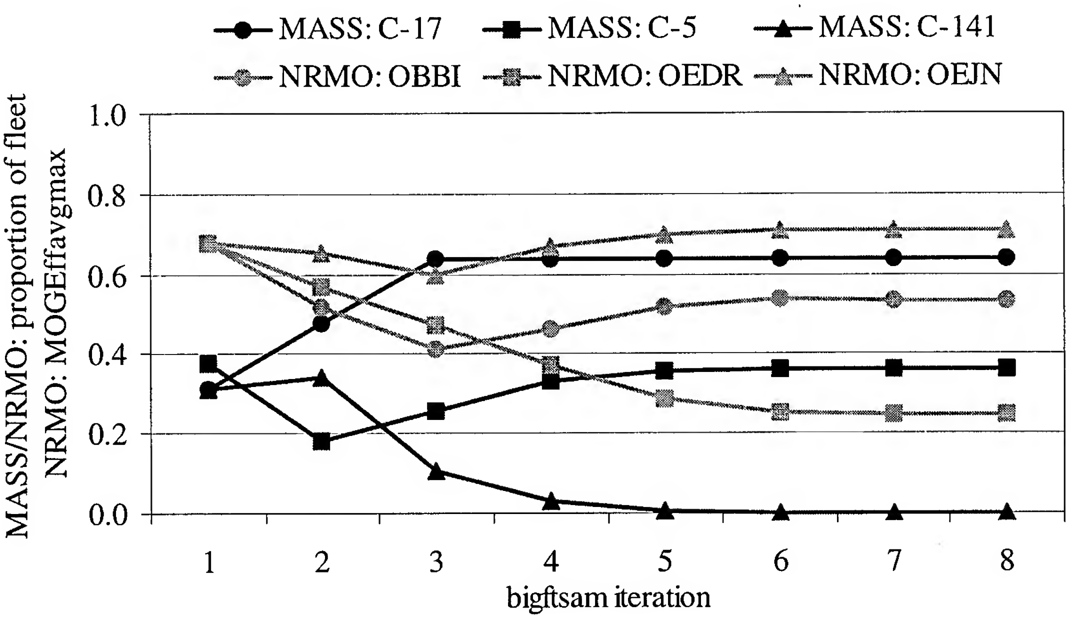

Figure 25: Input Convergence: bigftsam

Table 16 lists the total throughput of each model as well as the relative difference between MASS and NRMO for each throughput category. A point of interest in the table is that the relative difference between the models, while starting small (10.9 percent) in iteration 1, grows to 18.6 percent by the final iteration. This is due to MASS having use of more efficient aircraft which it flies continually without consideration of whether the requirement is satisfied or not. The only condition needed for MASS to attempt to move the next requirement in the TPFDD is that the requirement is available to load, as specified by its available-to-load date (ALD). NRMO, on the other hand, is also using the more efficient aircraft, and in fact is able to deliver about six percent more cargo in the last iteration than the first, but NRMO does not let itself deliver cargo early unless it would help its objective function. NRMO opts instead to allow aircraft to remain at home station and obtain the small bonus for doing so.

Table 16: Output Convergence: bigftsam

Iter.	Total stons/day		MASS is ___% greater than NRMO			
	MASS	NRMO	Bulk	Over	Out	Total
1	2085.4	1879.8	7.5%	5.9%	36.6%	10.9%
2	2038.6	1954.6	4.1%	7.3%	-2.9%	4.3%
3	2269.4	1982.7	11.8%	17.5%	11.3%	14.5%
4	2329.2	1983.7	16.6%	18.9%	15.0%	17.4%
5	2361.2	1980.0	16.9%	20.4%	20.6%	19.3%
6	2371.0	1991.2	15.6%	21.6%	19.2%	19.1%
7	2371.0	1999.2	20.7%	14.4%	26.9%	18.6%
8	2371.0	1999.2	20.7%	14.4%	26.9%	18.6%

Figure 26 graphically displays the convergence of the three throughput categories. The fact that the requirement is large enough to at least somewhat stress the airlift system, combined with the fact that the most efficient aircraft are allowed to be used, results in the most dynamic of these charts. These factors also allowed the total throughput for each model to increase with the iteration; the final MASS throughput growing almost fourteen percent from the first to the last iteration, and the final NRMO throughput growing over six percent.

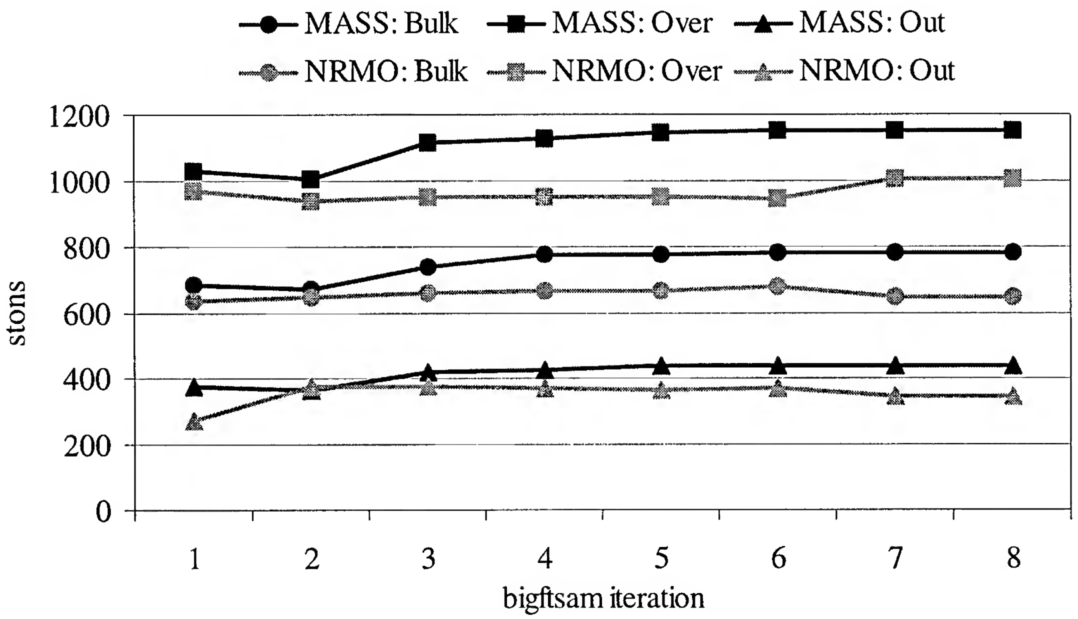

Figure 26: Output Convergence: bigftsam

Performance Results— Gradient Analysis

This section examines the performance of the design of experiments on the MASS and NRMO models with respect to the small and big scenarios. I use only the "ftsam" and "bigftsam" output/input crossflow applications to create experimental designs, since they represent the only applications in which the input sets for each model are as similar as possible. The input values from the final iteration of each output/input crossflow are used as the center point of each design. I run each experiment as listed in Table 4. Recall that the factors I vary to create this experimental design include the number of C-5 and C-17 aircraft and the amount of fuel available at each of the four en route bases, Ramstein, Germany; Mildenhall, England; Moron, Spain; and Rota, Spain. Thirty replications are run at each point in the MASS model to reduce variation. Of course, I run the NRMO model only once per design point since successive runs yield identical results. The constructed metamodels use the output total throughput as the dependent variable since it is a high level response of greatest concern to decision makers. If the models compare favorably using this output, other responses may be analyzed to determine the extent of covalidity. However, if the models do not compare favorably using this high level response, a strong claim may be made that the models are not covalid for the scenarios under observation. The first subsection details the experimental design performed on the small scenario, and the next subsection describes the experimental design applied to the big scenario.

Experimental Design—Small Scenario. The MASS model is run 30 times and NRMO is run once at each of the 65 design points. The initial regression metamodel is constructed using all six factors plus all two-way interactions. A lack of significance in the interaction terms combined with a lack of agreement of which interaction terms were significant led to dropping all the interactions and dealing only with the main factors. In this way, the metamodels for the MASS and NRMO models are readily comparable.

Table 17 lists a few statistics concerning the resulting metamodels for MASS and NRMO, built with regard to total throughput. (Note that the MASS metamodel is constructed using 65 × 30 = 1950 total runs while the NRMO model uses only 65 points.) The mean total throughput of the MASS model is 17.5 percent higher than that of the NRMO model, roughly as expected given the result of the output/input crossflow method (see Table 10).

Table 17: Small Scenario Total Throughput Metamodel Results

		MASS	NRMO
Coefficient	Mean	1616.4	1375.6
	C-5	-0.4	16.2
	C-17	0.9	2.5
	EDAR	-0.1	0.0
	EGUN	0.1	-0.3
	LEMO	0.5	-0.7
	LERT	0.1	2.0
P-Value	Mean	0.000	0.000
	C-5	0.013	0.000
	C-17	0.000	0.101
	EDAR	0.589	1.000
	EGUN	0.671	0.859
	LEMO	0.001	0.629
	LERT	0.736	0.187
Coefficient Standard Error		0.149	1.481
MSR		381.0	2918.9
MSE		42.6	140.3
F*		8.94	20.80
F* Significance		0.0	0.0
R^2		0.027	0.683

The only significant factors in the MASS metamodel are the aircraft (C-5 and C-17) and the fuel at LEMO (Moron). In the NRMO model, the number of C-5s is the only clearly significant factor, and the number of C-17s is marginally significant at the $\alpha =$ 0.10 level. That the aircraft factors could both be considered significant in each model is favorable to the consideration of the models' covalidity for this scenario until it is noticed that the signs of the coefficients do not agree for the C-5 factor. Of course, one would expect all the coefficient signs to be positive since an increase in any of the factor levels means an increase in resources available to deliver cargo. Specifically, the coefficients may be interpreted as the change in the total short tons delivered per day due to an increase in the factor level by approximately ten percent (eight aircraft for the aircraft factors, 62,250 gallons of fuel for each of the base fuel factors). That the signs on the MASS coefficients are not all positive, then, is a matter to look into more closely.

Recall that the scenario's movement requirements are not demanding for either model. MASS delivers requirements using the first available aircraft searching for the first available cargo requirement and carrying it using the first available route. Therefore, if the model has delivered all required cargo by the appropriate date, the model will still search for the next available piece of cargo and deliver it, effectively getting ahead on its required deliveries. The MASS model, however, can only pick-up cargo for delivery on or after the cargo available-to-load date (ALD). If MASS is picking up all or most of its cargo on the ALD, adding resources (or taking away resources to the extent that the model can still pick up all the cargo on the ALD) will not affect throughput. This is the case for the MASS model operating on the small scenario. The delivery requirement is easily handled regardless of small perturbations made to the resource levels studied, and the small coefficients are due to both random variation of the responses and nuances in the complex simulation model. A look at the R^2 indicates that the fitted model does not explain a significant portion of the variation of the responses, further demonstrating that the MASS regression metamodel is inadequate to explain changes in total throughput due to incremental changes in the factors of interest.

The NRMO regression metamodel displays, as the MASS metamodel, that the fuel at the en route bases is not a significant factor when considering this small scenario. The significance of the C-5 factor, however, allows the fitted NRMO metamodel to explain much more of the response variation than the MASS metamodel. The relatively large value for this factor is explained by the efficiency of the C-5 compared to the C-17 when fuel and MOG are not (tightly) constrained, as in the small scenario.

Experimental Design—Big Scenario. The big scenario is analyzed just as the small scenario. Thirty replications at each design point are run for the MASS design of experiments for a total of 1950 runs, while the NRMO metamodel is created using just 65 runs, one from each design point. Again, I construct the metamodels for total throughput using only the main effects in order to a linear gradient estimate.

Table 18 displays the characteristics of the MASS and NRMO metamodels created for the big scenario. The first point to note is that both metamodels account for a much larger percentage of the variations of the response, as indicated by the R^2 values, than in the small scenario. This is due to the fact that the scenario's increased size means that the models can no longer easily move the entire requirement in an environment that appears unconstrained. In particular, in the MASS metamodel, it is clear that the fuel available at the en route bases is becoming a constraint, considering the large coefficients of these factors. Interestingly, the NRMO model (delivering about 15 percent less cargo due to reasons discussed previously) seems to be on the edge of this fuel constraint, with only the fuel at Rota, Spain providing a significant factor. However, this is the base I would expect to notice the fuel constraint first since the initial amount of fuel available at Rota is less than at the other en route bases (250,000 gallons per day versus 800,000 gallons per day), while the amount I vary the fuel remains constant (62,250 gallons per day) regardless of the base.

Table 18: Big Scenario Total Throughput Metamodel Results

		MASS	NRMO
Coefficient	Mean	2341.4	1980.8
	C-5	2.3	16.2
	C-17	13.3	12.2
	EDAR	28.4	0.0
	EGUN	23.3	0.9
	LEMO	34.0	0.8
	LERT	34.1	12.9
P-Value	Mean	0.000	0.000
	C-5	0.009	0.000
	C-17	0.000	0.000
	EDAR	0.000	1.000
	EGUN	0.000	0.154
	LEMO	0.000	0.196
	LERT	0.000	0.000
Coefficient Standard Error		0.904	0.614
MSR		1231511.4	6199.8
MSE		1567.7	24.1
F*		785.55	257.15
F* Significance		0.0	0.0
R^2		0.708	0.964

In this big scenario, it is clear that MASS is unable to pick up the requirements on the ALD as in the small scenario. This is evidenced by the increased magnitude and significance of the coefficients. A comparison of the NRMO coefficients shows that the big scenario is more demanding, as well. The exception here is that fuel at the first three en route bases in the NRMO metamodel still has little significance in regards to throughput.

Therefore, whereas in the small scenario the metamodels are of limited significance, the big scenario metamodels appear much more conducive for comparison. In the following subsection, I discuss the comparison of the metamodels and allude to the implications of covalidity between the models.

Model Comparison. As discussed in Chapter 3, I consider two methods to assist in the determination of models' covalidity. The first method of establishing the significance of a "difference" model was exercised on the test models of Chapter 3. The other method considers whether one of the metamodel's gradient vectors established in the preceding subsections can be found within the confidence region (hypercone) of the other metamodel's gradient vector. Of course, this method tests only the similarity of the gradient values and should be accompanied by an examination of the differences between the mean response values (unless an application is solely concerned with the sensitivity of the response to the inputs).

I treat the small scenario comparison first. Of course, the difference between the mean values for the total throughput output of the two metamodels is statistically non-zero, so I concentrate here on the analysis of the gradient values. The angle between the gradient vector and the extent of the confidence hypercone has been described by Equation (25). I construct a 95 percent confidence hypercone about the MASS small scenario metamodel gradient vector and determine that the confidence hypercone extends 27.05 degrees from the gradient vector. The angle between the MASS and NRMO metamodel gradient vectors is determined by the taking the arc cosine of the dot product of the vectors divided by the product of the magnitude of each vector. I calculate the angle between the gradient vectors to be 103.04 degrees. This angle being larger than a right angle implies that at least one factor's coefficient is of opposite sign from one model to the other. I noted this phenomena in the discussion of the small scenario's C-5 factor. The gradient comparison results are summarized in Table 19.

Table 19: Gradient Comparison Summary

Scenario	Angle Between Gradients	95% Confidence Region Angle	
		MASS	NRMO
Small	103.04 °	27.05 °	17.96 °
Big	62.61 °	2.78 °	5.03 °

I acknowledge that there is a degree of latitude when making the determination as to whether models are covalid or not. For instance, determining the F statistic in Equation (25) to a different level of significance would give a different angle to test the gradient vectors against (finding a 99 percent confidence region for the small scenario results in an angle that extends 32.1 degrees from the MASS metamodel gradient vector). No confidence region may extend beyond 90 degrees from the gradient vector, however, since the angle is found by taking an arc sine. So I may clearly state that these metamodels are not close enough in terms of response or gradients developed with respect to inputs of interest to claim that the MASS and NRMO models are covalid at the input point identified as the small scenario.

Next, I examine the big scenario. Again, the difference between the mean total throughput values for the metamodels created for the big scenario is so great that the means cannot be considered equal. The angle between the gradient vector of the MASS model's metamodel and the extent of its 95 percent confidence region hypercone works out to only 2.78 degrees. In an examination of Equation (25), I recognize that this is due to the large magnitudes of the gradient coefficients. The angle between the MASS and NRMO metamodel gradient vectors is found to be 62.6 degrees.

Obviously, the NRMO metamodel gradient vector does not exist inside the confidence region of the MASS metamodel's gradient vector. I also consider whether a confidence region constructed around the NRMO metamodel gradient vector would overlap the MASS metamodel gradient vector confidence region. The confidence region created around the NRMO metamodel gradient vector extends 5.03 degrees from the vector. Therefore, the 95 percent confidence regions for the two metamodels' gradient vectors do not overlap.

Though I cannot claim that the MASS and NRMO models compare favorably enough at the small or big scenarios to warrant the claim that they are covalid, some consolation may be taken in the fact that the coefficients from the big scenario's metamodels all agree in sign. This information may be used by system experts to claim some limited level of covalidity between the models at the big scenario, particularly since the signs of the gradient coefficients all agree with expectations, namely that an increase in resources will yield an increase in total throughput. Also, the fact that for the big scenario gradient vectors are closer than in the small scenario could lead experts to consider that an even larger, more constrained scenario may produce closer gradient vectors. This consideration could lead to the creation of such a scenario, and the entire output/input crossflow method and gradient analysis could be performed at this new scenario.

V. Conclusions and Suggested Future Research

Synopsis

I have devised a method that allows for comparisons between models under different modeling paradigms. I began with the description of a general modeling paradigm, particularly applicable when multiple models are created to represent the same system. The paradigm serves as a framework for the developed methodology. My proposal is to use two or more models in concert in order to improve the performance of each. The output/input crossflow method is developed to generate this model improvement. Through the use of the output/input crossflow method, we hope to arrive at models that are as close as possible in terms of their inputs and outputs. I then develop a method to characterize the difference not only in the model outputs, but also in the models' sensitivities to input variations. The end result is a characterization of the models' covalidity. I apply the developed methods both to a pair of small-scale test models and to two large-scale models used by AMCSAF to analyze the strategic airlift system, MASS, a simulation model, and NRMO, an optimization model. This chapter summarizes the research in each of these areas, offers suggestions for further research in this area, and finally provides conclusions.

General Modeling Paradigm

In defining the general modeling paradigm, I describe mathematically the creation of models as they are drawn from the real world. I begin by developing a mathematical idiom describing the real world. From this I circumscribe a system and describe the way reality transforms this real-world system from an input state to an output state through some "truth" concerning the real-world system. A model is merely an attempt to accurately aggregate this circumscription into a mathematical form. The input to a model is an attempt to aggregate the state of the system at a defined start time, and the output of the model is intended to accurately describe the system at the defined termination time. I assert, then, that a model is a mapping from the input to the output. I acknowledge that any number of models may be constructed to represent the same system, though each model usually requires its own input and output set due to the paradigm (i.e., simulation or optimization) and assumptions under which each model is built.

Models which map their inputs to outputs as designed in light of their corresponding real-world system are "valid" models. Often, however, it is not practical to perform a comparison between a model and its real-world system, thereby limiting the ability to validate the model. If two or more models are created to represent the same real-world system, however, a comparison may be performed between the models. Models that compare favorably, one to another, are called "covalid" models. Models that merely compare favorably to one another can be termed "covalid in the narrow sense," while such models, if they should also be determined to compare favorably to the real world, may be called "covalid in the wide sense." If one of two covalid models has been independently validated, we may claim the other model is "valid by association" with respect to those areas of comparison.

Output/Input Crossflow

In the quest to determine the covalidity of two (or more) models, I acknowledge that certain assumptions required or inherent in one model may not match those of another model. In an attempt to compare models that as nearly as possible resemble one another in terms of inputs, I develop the method of output/input crossflow. This method seeks to improve the inputs of one model based on the output of another (or others). This iterative method requires the user to find feedback functions that translate subsets of output from one model into inputs for the other model. Input variables that have potential for this type of update include parameters whose initial values have little justification and variables that one of the models outputs as "optimum." I prove that there exist feedback functions that allow this iterative method to converge at fixed points (regarding the parameters and variables used in the crossflow). The resultant input sets are considered to represent the represented scenario as nearly as possible.

I exercised the output/input crossflow method on two test models. The inputs (and outputs) converged after 11 iterations. Further, the difference in the relevant output metric for the two models decreased (by the last iteration) by 80 percent compared to the initial difference between the models.

I also applied the method to the large-scale MASS simulation and NRMO optimization models employed by AMCSAF. Using three different sets of crossflow functions on two scenarios each, I achieve convergence in ten iterations or less in every case. The results of the attained fixed points provide a more efficient mix of military aircraft as used by the MASS model and base-specific values for the efficiency of MOG usage in NRMO, values that previously had no empirical justification.

Gradient Analysis

With the input values considered as close as possible, I developed a method to compare models that not only considers the mean response at a scenario, but also the gradient of the response with respect to specified inputs of interest. I suggest a factorial design of experiments with design points no more than ten percent away from the center point to approximate the gradient, though I appreciate that there are other methods available to approximate these gradients. Linear metamodels are constructed about the design center point and the resulting coefficients correspond to the gradient of the response with respect to each input.

I describe two methods for comparing the resultant metamodels. First, I take the point-by-point differences between the models' responses and create a metamodel from these differences. If the models describe the system identically, the created metamodel would be a "zero" model with no significant coefficients (or mean value). The extent of the "difference" model's significance, then, is an indication of a lack of covalidity between the models. The degree to which the "difference" model must be a "zero" model, however, is a judgement required in each application and may vary.

The second method creates the linear metamodels as described, and compares the attributes of these metamodels. A confidence region is created about the gradient vector. An angle represents the extent of this confidence region from the gradient vector. The angle between the gradient vectors is calculated, and if this angle is less than the angle representing the confidence region, the models are declared covalid.

A gradient analysis was performed on the test models using the "difference" model method. While the result was a significant model (implying the models are not covalid), the “difference” model accounted for only a small part of the variance of the responses. System experts are required, then, to determine exactly how close models should be before they are labeled covalid. This determination should take into account the type of models involved as well as the intended use of each model.

I have also performed a gradient analysis on the small and big scenarios involving the MASS and NRMO models using the confidence region comparison method. Since the models’ gradient vectors were significantly different in each scenario, the models are not considered covalid at the scenarios employed. However, insights from the method point to possible scenarios where the models could compare better.

Future Research

I have developed a framework about which models representing various real-world systems may be constructed and covalidated or validated. An obvious first extension to this research is to construct further MASS and NRMO scenarios with TPFDDs for which the models would compare more favorably. I have shown that the models may not be considered covalid at the two chosen scenario points, but the research also suggests regions in the scenario space where the models outputs could be closer than in my examples.

One area of future research that is of interest would be to use the developed methods on three or more models. For instance, a model designed to simulate airfield capabilities could be used in concert with the MASS and NRMO models to provide even better insights to these models concerning MOG use. A required extension for carrying out the gradient analysis portion of the method would be the development of a method for performing three-way comparisons between models.

Another promising use for this research would be to use the developed methods for covalidating follow-on models of established, legacy models. Either in newer versions of existing models or for actual replacement models, the developed methods provide a logical means of determining the reasonableness of using the newer model. Further, the methods here would be appropriate for the testing of new functionality added to existing models.

Another contribution would be to develop a method to determine the entire region of scenario space over which two models are covalid. The state of being covalid (or better yet, valid) at one or two scenario points would go a long way in terms of developing trust (or accreditation) in the models over a range of inputs. However, one can not generalize the entire scenario space, or even a region of it, based on one or two points. As alluded to, I acknowledge that multiple methods exist for performing the comparison of gradient metamodels. Further work is required to detail these. It is considered useful to develop methods whereby a determination may be made as to the extent of covalidation, similar to determining the angle between gradient vectors as done here.

One could evaluate a metric developed to represent the extent of covalidity between models over an experimental design that spans a significant region of the scenario space. The ensuing response surface would point to regions where the models are “most” covalid (particularly if non-linear metamodels are created) and also to other regions (outside the design space) where the models may compare favorably. In this way, designs could be used both to explore areas of the scenario space that meet some established criteria of covalidness and as a means of finding the direction of steepest descent to an area of the scenario space that is “more” covalid.

Another area which requires further exploration is the treatment of the real-world system in question as the “truth” model. Performing the output/input crossflow method on a “truth” model and an optimization model, for instance, could lead to process improvements in the actual system. Performing the method of gradient analysis on the optimization model (or multiple models) and the “truth” model provides an innovative technique to validate the model(s).

Conclusion

I have effectively established an original paradigm regarding the construction, improvement, and validation of models that represent real-world systems. This paradigm describes the translation of real-world systems to the modeling world, the way in which models (and the real-world system) may be enhanced based on the findings of the other models (or the system itself), and methods of determining the correctness of the models, both in relation to each other and to the real-world system they attempt to characterize.

This all-encompassing look at the breadth of modeling, from the real-world system to the validation of the model can be useful at any level of model creation. While each application of this methodology may occur to a different extent, the general paradigm may be followed no matter how complex the system or the model. What is more, varying complexities of the constructed models are not an issue, either. It is my hope that this work may put the creation of future models into a slightly different perspective than in the past.

First, models have a relation to the real-world system they attempt to represent. Ideally, the model's input, output, and function all draw from this real-world system. In areas where it is unpractical or impossible to take information directly from the real-world system, the modeler makes assumptions and should realize that these assumptions may negatively impact the performance of the model.

Second, instead of considering two models created for the same real-world system as rivals, they may be considered allies. Developing multiple perspectives of the same divisive topic tends to create a clarity in the consideration of the topic. In the same way, multiple models developed for the same system can create a better understanding of the system that leads to improvements in both models. By realizing these improvements, the hope is to decrease the impact that assumptions have on the models.

Third, the comparison of models with the real world has long been known to lead to increased faith and trust in models. I argue that the comparison of models with each other leads to faith and trust that the models are mimicking the real world in the same way. This covalidity in the narrow sense may be expanded to a covalidity in the wide sense if the models also compare favorably with the real-world system. The important note here is that models should have some benchmark of comparison if they are to be used in rigorous analysis, and if the real-world system upon which they are based is not available for that use, another model can serve as a reasonable proxy. The extent of these models' believability relies on this comparison.

Further, the entire methodology was performed on two large-scale, real-world models employed by the United States Air Force Air Mobility Command, the MASS simulation and the NRMO linear program. While the conclusion is that the models are not covalid in terms of the scenarios analyzed, the research has illuminated several insights into the operations of these models and points to areas of the scenario space where the models may compare more favorably.

Appendix A. Test Models and Feedback Functions

Figure 27 shows the basic network used in the test models. In the network for the base case (center of experimental design), 50 aircraft are sent from Home to either base A or base B. At base A or base B, the aircraft are unloaded and serviced, and they return to Home. Scenarios for both the simulation and the optimization cover 15 days. The flight and service time distributions shown are used in the simulation, but the optimization formulation only reflects the mean times.

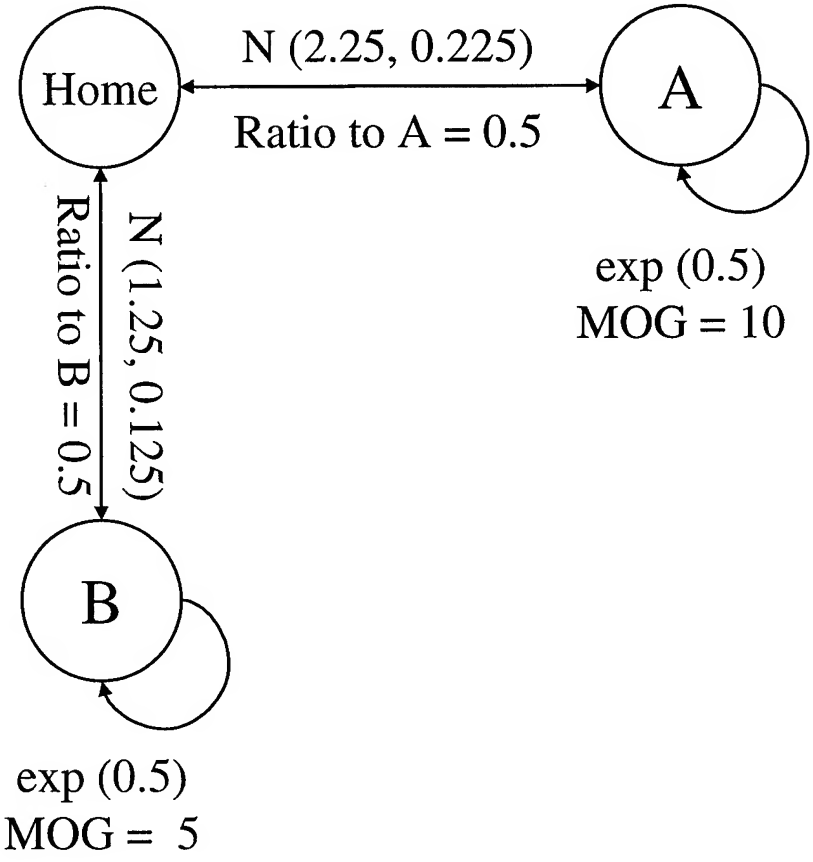

Figure 27: Test Model Network

The optimization formulation is shown below. The basic formulation maximizes throughput. Constraints are added which limit the number of planes flown each day to 50 and the number of planes serviced by a base to some fraction (MOG efficiency) of the available MOG at the base. Not shown is an additional constraint that accounts for the amount of time required for the initial aircraft to reach the bases.

maximize $thruput = \sum_{t} \sum_{b} \mathbf{X}(b,t)$

subject to:

$$\sum_{b} \sum_{flying(t)} \mathbf{X}(b, flying(t)) \leq plane \quad \forall t$$

$$\mathbf{X}(b,t) \leq MOGeff \times MOG(b) \quad \forall b,t$$

$$\mathbf{X}(b,t) \geq 0 \quad \forall b,t$$

where

b is base A or B

t is day 1 through 15

X is number of missions arriving at *b* during day *t*

flying is a vector which accounts for the flight days during a mission to *b* arriving at *t*

plane is total number of aircraft available

MOG is the maximum number of aircraft simultaneously serviceable by *b*

MOGeff is a measure of how efficiently *MOG* can be used if *b* is bottlenecked

The equations that derive the input of one test model from the output of the other are specified below. The equation which filters Baby MASS output into Baby NRMO input calculates the fraction of MOG which the simulation could actually use at a base, given that the base was bottlenecked throughout the simulation, and is given as Equation (30).

$$\begin{aligned} \text{MOG eff.} &= \frac{\text{\# missions}}{\text{\# days}} \div \text{avail. MOG / day} \\ &= \text{MOGeff missions / MOG} \end{aligned} \tag{30}$$

The equation which filters Baby NRMO output to Baby MASS input determines the proportion of missions the Baby NRMO optimization flies to each base (A and B), and is presented as Equation (31). As a function of total throughput to a base over the entire fifteen-day scenario, missions flown per day is determined by how many days to which each base is flown. In this scenario, base A was flown to for 13 of the 15 days, and base B was flown to for 14 days.

$$\frac{A}{B} = \frac{\text{Total to A/\# days flown to A}}{\text{Total to B/\# days flown to B}} \tag{31}$$

Appendix B. Pertinent MASS and NRMO Input

This appendix displays some pertinent input files used by the MASS and NRMO models for the small and big scenarios. With the exception of the TPFDD file, all files for the small and big scenarios are identical. Each input file has been truncated to show only information required to understand the application, and some fields have been omitted as a result.

Table 20 shows the list of locations used by the scenario. The first column lists the ICAO; the first two columns of numbers list the narrow-body and wide-body MOG, respectively (connected by an "or" statement); the next column lists the gallons of fuel available at the base each day; and the last column lists the common name of the airfield. Note that a field with all "9"s effectively provides an unconstrained amount of the resource in question.

Table 20: Scenario Location List

```
EDAR    9 or    4  800000 RAMSTEIN AB
EGUN    4 or    2  800000 MILDENHALL
LEMD   28 or   14 9999999 BARAJAS
LEMO    4 or    4  800000 MORON AB
LERT    2 or    1  250000 ROTA NS
EDDF 9999 or 9999 9999999 FRANKFURT MAIN
EGLL   32 or   16 9999999 HEATHROW
KCHS 9999 or 9999 9999999 CHARLESTON AFB/MUNI
KJFK 9999 or 9999 9999999 JOHN F KENNEDY INTL
KWRI 9999 or 9999 9999999 MCGUIRE AFB
OBBI   26 or   13 9999999 BAHRAIN INTL
OEDR   10 or    5 9999999 DHAHRAN INTL
OEJN   13 or    6 9999999 KING ABDUL AZIZ INTL
```

Appendix B

This appendix displays some pertinent input files used by the MASS and NRMO models for the small and big scenarios. With the exception of the TPFDD file, all files for the small and big scenarios are identical. Each input file has been truncated to show only information required to understand the application, and some fields have been omitted as a result.

Table 20 shows the list of locations used by the scenario. The first column lists the ICAO; the first two columns of numbers list the narrow-body and wide-body MOG, respectively (connected by an "or" statement); the next column lists the gallons of fuel available at the base each day; and the last column lists the common name of the airfield. Note that a field with all "9"s effectively provides an unconstrained amount of the resource in question.

Table 20: Scenario Location List

```
EDAR    9 or    4  800000 RAMSTEIN AB
EGUN    4 or    2  800000 MILDENHALL
LEMD   28 or   14 9999999 BARAJAS
LEMO    4 or    4  800000 MORON AB
LERT    2 or    1  250000 ROTA NS
EDDF 9999 or 9999 9999999 FRANKFURT MAIN
EGLL   32 or   16 9999999 HEATHROW
KCHS 9999 or 9999 9999999 CHARLESTON AFB/MUNI
KJFK 9999 or 9999 9999999 JOHN F KENNEDY INTL
KWRI 9999 or 9999 9999999 MCGUIRE AFB
OBBI   26 or   13 9999999 BAHRAIN INTL
OEDR   10 or    5 9999999 DHAHRAN INTL
OEJN   13 or    6 9999999 KING ABDUL AZIZ INTL
```

Table 21 shows when and where aircraft are introduced into the scenario. The first column lists the aircraft type (wbp stands for wide-body passenger aircraft); the second column shows the ICAO of the base at which the aircraft are introduced; the third column lists the scenario day the aircraft are introduced; the fourth column shows how many of that type aircraft are introduced; the fifth column shows the cumulative number of aircraft of that type in the scenario; and the last column shows the average number of hours per day each aircraft introduced may fly. The number of hours per day represents the surge utilization rate that is applied for the first 45 days of a conflict. The wbp utilization rate is set by contract.

Table 21: Scenario Aircraft Use

WBP	KJFK	0	25	25	12.0
C-141B	KWRI	0	20	20	12.2
C-141B	KCHS	0	10	30	12.2
C-141B	KWRI	1	15	45	12.2
C-141B	KCHS	2	5	50	12.2
C-5	KWRI	0	15	15	10.7
C-5	KCHS	0	15	30	10.7
C-5	KWRI	1	15	45	10.7
C-5	KCHS	2	15	60	10.7
C-17	KCHS	0	30	30	15.3
C-17	KCHS	1	15	45	15.3
C-17	KCHS	2	5	50	15.3

Table 22 displays the list of requirements, or the TPFDD, used in the small scenario. The first column is the simply a sequential requirement identifier; the second column lists the available-to-load date, on or after which the requirement is able to be picked up; the third column lists the required delivery date, on or before which the requirement must be delivered; the two columns of ICAOs list the onload location and the offload location, respectively; the following three columns list the outsize, oversize, and bulk short tons to be moved; and the last column lists the number of passengers required to be moved.

Table 22: Small Scenario Requirements List (TPFDD)

1	0	4	KWRI	OEJN	0.0	8.0	5.0	135
2	0	4	KCHS	EDAR	0.0	2.0	13.0	22
3	0	4	KWRI	OBBI	20.0	0.0	25.0	55
4	0	4	KCHS	OEDR	32.0	42.0	12.0	272
5	1	4	KWRI	OEDR	17.0	8.0	5.0	35
6	1	4	EDAR	OEJN	0.0	12.0	13.0	142
7	1	5	KCHS	OEJN	2.0	90.0	25.0	55
8	1	5	LEMO	OBBI	36.0	55.0	12.0	122
9	1	5	KCHS	OEJN	2.0	90.0	25.0	355
10	1	6	EDAR	OBBI	0.0	62.0	36.0	92
11	1	6	KWRI	OEDR	48.0	227.0	86.0	195
12	2	6	EDAR	KWRI	0.0	2.0	4.0	35
13	2	6	EDAR	OEDR	0.0	12.0	13.0	242
14	2	6	KCHS	OEDR	2.0	90.0	25.0	55
15	2	6	LEMO	OEDR	29.0	47.0	1.0	184
16	1	6	KCHS	OEJN	357.0	524.0	272.0	870
17	2	6	EDAR	OBBI	0.0	28.0	41.0	16
18	2	6	KWRI	OEJN	30.0	52.0	38.0	321
19	2	6	KCHS	OEJN	10.0	61.0	30.0	135
20	3	6	KCHS	EDAR	19.0	42.0	13.0	22
21	3	7	KWRI	OBBI	100.0	0.0	25.0	74
22	3	7	KWRI	OEDR	32.0	55.0	3.0	72
23	2	7	KWRI	OEDR	257.0	487.0	998.0	468
24	3	7	EDAR	OEJN	3.0	14.0	22.0	20
25	3	7	KCHS	OBBI	7.0	0.0	1.0	13
26	3	7	KWRI	OEJN	96.0	535.0	312.0	122
27	3	7	KCHS	OEJN	2.0	90.0	25.0	355
28	3	7	KCHS	OBBI	0.0	19.0	36.0	92
29	3	7	KWRI	OEDR	19.0	127.0	66.0	95
30	4	7	EDAR	KWRI	0.0	2.0	4.0	35
31	4	8	EDAR	OEDR	0.0	12.0	13.0	342
32	4	8	KCHS	OEJN	2.0	90.0	25.0	55
33	4	8	KCHS	OBBI	85.0	247.0	80.0	184
34	4	8	EDDF	OEJN	15.0	24.0	72.0	270
35	4	8	KWRI	OBBI	23.0	41.0	127.0	16
36	4	8	KWRI	OEJN	30.0	52.0	38.0	154
37	4	8	KWRI	OBBI	0.0	22.0	89.0	208
38	4	8	EDAR	OBBI	0.0	5.0	3.0	325
39	4	9	KCHS	OEJN	99.0	424.0	280.0	174
40	5	9	KWRI	OEJN	15.0	45.0	14.0	69
41	5	9	KCHS	OEDR	21.0	80.0	48.0	468
42	5	9	EDAR	OEDR	3.0	14.0	22.0	20
43	5	9	LEMO	KCHS	7.0	16.0	1.0	69
44	5	9	KCHS	OBBI	16.0	235.0	12.0	122
45	5	9	KCHS	OEJN	2.0	90.0	25.0	355

46	5	10	KCHS	OBBI	0.0	19.0	36.0	92
47	5	10	KWRI	OEDR	19.0	127.0	66.0	95
48	5	10	EDAR	KWRI	0.0	6.0	8.0	21
49	2	10	KJFK	OEDR	0.0	12.0	548.0	2249
50	6	10	KWRI	OBBI	0.0	101.0	88.0	55
51	6	11	KCHS	OEDR	94.0	127.0	236.0	348
52	3	11	KJFK	OEJN	9.0	24.0	98.0	1172
53	6	11	KCHS	OBBI	219.0	397.0	164.0	64
54	6	11	KCHS	OEDR	24.0	18.0	62.0	239
55	4	11	KWRI	OEJN	241.0	540.0	119.0	235
56	4	11	KCHS	EDAR	90.0	478.0	213.0	622
57	6	11	KWRI	OBBI	20.0	0.0	25.0	355
58	7	11	KCHS	OEDR	32.0	42.0	12.0	72
59	7	11	KWRI	OEDR	57.0	98.0	65.0	235
60	7	12	EDAR	OEJN	0.0	27.0	17.0	142
61	5	12	KCHS	OEJN	172.0	370.0	159.0	1055
62	7	12	LEMO	OBBI	36.0	55.0	12.0	322
63	7	12	KCHS	OEJN	352.0	690.0	325.0	355
64	7	12	EDAR	OBBI	0.0	62.0	36.0	292
65	7	12	KWRI	OEDR	38.0	253.0	136.0	195
66	8	12	EDAR	OEDR	0.0	12.0	13.0	42
67	8	13	KCHS	OEDR	2.0	90.0	25.0	55
68	8	13	LEMO	OEDR	29.0	47.0	1.0	184
69	8	13	KCHS	OEJN	15.0	24.0	72.0	270
70	8	13	EDAR	OBBI	0.0	28.0	41.0	16
71	8	13	KWRI	OEJN	30.0	52.0	38.0	291
72	8	13	KCHS	OEJN	10.0	61.0	30.0	135
73	9	13	KCHS	EDAR	74.0	84.0	37.0	522
74	9	13	KWRI	OBBI	100.0	0.0	25.0	674
75	9	14	KWRI	OEDR	32.0	55.0	3.0	272
76	9	14	KWRI	OEDR	17.0	48.0	15.0	468
77	9	14	EDAR	OEJN	3.0	14.0	22.0	420
78	9	14	KCHS	OBBI	47.0	90.0	51.0	13
79	9	14	KWRI	OEJN	16.0	235.0	12.0	222
80	10	14	KCHS	OEJN	2.0	90.0	25.0	355
81	10	15	KCHS	OBBI	0.0	19.0	36.0	92
82	10	15	KWRI	OEDR	19.0	127.0	66.0	495
83	10	15	EDAR	KWRI	0.0	2.0	4.0	35
84	10	15	EDAR	OEDR	0.0	12.0	13.0	342
85	10	15	KCHS	OEJN	2.0	90.0	25.0	155
86	10	15	KCHS	OBBI	29.0	47.0	1.0	184
87	11	15	EDDF	OEJN	15.0	24.0	72.0	270
88	11	15	KWRI	OBBI	23.0	41.0	127.0	16
89	11	16	KWRI	OEJN	30.0	52.0	38.0	1154
90	11	16	KWRI	OBBI	0.0	22.0	89.0	208
91	11	16	EDAR	OBBI	0.0	5.0	3.0	325
92	11	16	KCHS	OEJN	71.0	224.0	0.0	174
93	11	16	KWRI	OEJN	15.0	45.0	14.0	369
94	12	16	KCHS	OEDR	21.0	80.0	48.0	468
95	12	17	EDAR	OEDR	3.0	14.0	22.0	20
96	12	17	LEMO	KCHS	7.0	0.0	1.0	13
97	12	17	KCHS	OBBI	16.0	235.0	12.0	122
98	12	17	KCHS	OEJN	2.0	90.0	25.0	355
99	12	17	KCHS	OBBI	0.0	19.0	36.0	92
100	12	18	KWRI	OEDR	19.0	127.0	66.0	95

```
101 13 18 EDAR KWRI   0.0   6.0   8.0   21
102 13 18 KJFK OEDR   0.0  12.0  48.0 2242
103 13 18 KWRI OBBI   0.0 101.0  88.0   55
104 13 18 KCHS OEDR  14.0  27.0  36.0   48
105 13 18 KJFK OEJN  79.0  84.0  38.0  462
106 13 18 KCHS OBBI 214.0 397.0 264.0  564
107 10 19 KCHS OEDR 424.0 818.0 762.0 1629
108 13 16 KWRI OBBI 121.0 280.0 148.0  468
109 13 17 EDAR KWRI  23.0  37.0  53.0   20
110 14 17 LEMO OBBI   7.0   0.0   1.0   13
111 14 17 KWRI OEJN 210.0 235.0  12.0  122
112 14 17 KCHS OEDR   2.0  90.0  25.0  355
113 14 17 KCHS OBBI   0.0  19.0  36.0   92
114 14 18 KWRI OEDR  19.0 127.0  66.0   95
115 15 18 EDAR KWRI   0.0   6.0   8.0   21
116 15 18 KJFK OEJN   0.0  12.0  48.0 2242
117 15 18 KCHS OEJN  40.0 351.0 103.0  255
118 15 18 KCHS OEDR  14.0  27.0  36.0   48
119 15 19 KJFK OEJN   9.0  24.0   8.0  462
120 15 19 KCHS OBBI   4.0  97.0  64.0   64
121 12 19 KCHS OEDR  24.0  18.0  62.0 1629
122 15 20 KCHS OBBI   4.0  97.0  64.0   64
123  8 12 EDAR KWRI   0.0   2.0   4.0   35
```

The TPFDD file used for the big scenario is identical to that of Figure 30, except each requirement (outsize, oversize, bulk, and passenger) is multiplied by 1.5 and rounded to the nearest tenth of a short ton (for cargo requirements) or the nearest whole passenger. All other input files for the big scenario are identical to those of the small scenario.

Bibliography

Adelman, Howard M. "Experimental Validation of Structural Optimization Methods," *NASA Technical Memorandum 104203*: (January 1992).

Aigner, Dennis J. "A Note on Verification of Computer Simulation Models," *Management Science, 18*: 615-619 (July 1972).

Apostol, Tom. *Mathematical Analysis, 2nd ed.* Reading MA: Addison-Wesley, 1974.

Arthur, James D. and Richard E. Nance. "Independent Verification and Validation: A Missing Link in Simulation Methodology," *Proceedings of the 1996 Winter Simulation Conference*: 230-236 (1996).

Balci, Osman. "Validation, Verification, and Testing Techniques throughout the Life Cycle of a Simulation Study," *Annals of Operations Research, 53*: 121-173 (1994).

Bermant, A. F. *A Course of Mathematical Analysis, Part I.* New York: MacMillan, 1963.

Border, Kim C. *Fixed Point Theorems with Applications to Economics and Game Theory.* Cambridge: Cambridge University Press, 1985.

Box, George E. P. and Norman R. Draper. *Empirical Model-Building and Response Surfaces.* New York: John Wiley & Sons, 1987.

Box, G. E. P. and W. J. Hill. "Discrimination among Mechanistic Models," *Technometrics, 9*: 57-71 (February 1967).

Box, G. E. P. and William G. Hunter. "A Useful Method for Model Building," *Technometrics, 4*: 301-318 (August 1962).

Box, George E. P., William G. Hunter, and J. Stuart Hunter. *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building.* New York: John Wiley & Sons, 1978.

Callahan, John and George Sabolish. "A Process Improvement Model for Software Verification and Validation," *Proceedings of the 19th Annual Software Engineering Workshop (NASA-CR-189411)*: 151-171 (December 1994).

Cavitt, David B., C. Michael Overstreet, and Kurt J. Maly. "A Performance Analysis Model for Distributed Simulations," *Proceedings of the 1996 Winter Simulation Conference*: 629-636 (1996).

Cohen, Kalman J. and Richard M. Cyert. "Computer Models in Dynamic Economics," *Quarterly Journal of Economics, 75*: 112-127 (February 1961).

Deslandres, V. and H. Pierreval. "An Expert System Prototype Assisting the Statistical Validation of Simulation Models," *Simulation, 56*: 79-89 (February 1991).

Diener, D.A., H.R. Hicks, and L.L. Long. "Comparison of Models: Ex Post Facto Validation/Acceptance," *Proceedings of the 1992 Winter Simulation Conference*: 1095-1103 (1992).

Elmer, Michael R. *Issues and Challenges in Validating Military Simulation Models.* MS Thesis, AFIT/GSO/ENS/95D-02. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1995 (AD-A303046).

Friedman, Linda W. and Israel Pressman. "The Metamodel in Simulation Analysis; Can It Be Trusted?," *Journal of the Operational Research Society, 39*: 939-948 (1988).

Hunter, William G. and Albey M. Reiner. "Designs for Discriminating between Two Rival Models," *Technometrics, 7*: 307-323 (August 1965).

Istratescu, Vasile I. *Fixed Point Theory: An Introduction*, Dordrecht Holland: D. Reidel Publishing Company, 1981.

Johnson, K. E., K. W. Bauer, Jr., J. T. Moore, and M. Grant. "Metamodelling Techniques in Multidimensional Optimality Analysis for Linear Programming," *Mathematical and Computer Modelling, 23*: 45-60 (1996).

Kleijnen, Jack P. C. "Statistical Validation of Simulation Models," *European Journal of Operations Research, 87*: 21-34 (1995a).

Kleijnen, Jack P.C. "Verification and Validation of Simulation Models," *European Journal of Operations Research, 82*: 145-162 (1995b).

Kleijnen, Jack P.C. "Five-Stage Procedure for the Evaluation of Simulation Models through Statistical Techniques," *Proceedings of the 1996 Winter Simulation Conference*: 248-254 (1996).

Kleijnen, Jack P.C., Bert Bettonvil, and Wilhelm Van Groenendaal. "Validation of Trace-Driven Simulation Models: Regression Analysis Revisited," *Proceedings of the 1996 Winter Simulation Conference*: 352-359 (1996).

Kleijnen, Jack P.C. and Wilhelm Van Groenendaal. *Simulation: A Statistical Perspective.* New York: John Wiley & Sons, 1992.

Law, Averill M. and W. David Kelton. *Simulation Modeling & Analysis, 2nd ed.* New York: McGraw-Hill, 1991.

Lee, Hau L. and Corey Billington. "Managing Supply Chain Inventory: Pitfalls and Opportunities," *Sloan Management Review, 33*: 65-73 (Spring 1992).

McCanne, Randy. *The Airlift Capabilities Estimation Prototype: A Case Study in Model Validation*. MS Thesis, AFIT/GOR/ENS/93M-13. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1993 (AD-A262603).

Merrill, David L. Analyst, Air Mobility Command Studies and Analysis Flight. Point paper on the Mobility Analysis Support System prepared for Congressional Budget Office analysts. Air Mobility Command, Scott AFB IL, 27 October 1993.

Montgomery, Douglas C. *Introduction to Statistical Quality Control, 2nd ed.* New York: John Wiley & Sons, 1991.

Morton, David P., Richard E. Rosenthal, and Lim Teo Weng. "Optimization Modeling for Airlift Mobility," *Military Operations Research,* 1: 49-67 (Winter 1996).

Neter, John, William Wasserman, and Michael H. Kunter. *Applied Linear Statistical Models, 3rd ed.* Homewood IL: Richard D. Irwin, 1990.

Pritsker, A. Alan B. *Introduction to Simulation and SLAM II, 3rd ed.* New York: John Wiley & Sons, 1986.

Randolph, John F. *Basic Real and Abstract Analysis*. New York: Academic Press, 1968.

Rousseau, Glenn G. *Airlift System Sensitivity to Perturbed Time-Phased Force Deployment Data*. MS Thesis, AFIT/GOA/ENS/96M-07. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 1996 (AD-A324266).

Rousseau, Glenn G. and Kenneth W. Bauer, Jr. "Sensitivity Analysis of a Large Scale Transportation Simulation Using Design of Experiments and Factor Analysis," *Proceedings of the 1996 Winter Simulation Conference*: 1426-1432 (1996).

Sanchez, Susan M., L. Douglas Smith, and Edward C. Lawrence. "Sensitivity and Scenario Analysis for Simulation Metamodels," *Proceedings of the 1996 Winter Simulation Conference*: 1440-1447 (1996).

Sargent, Robert G. "Some Subjective Validation Methods Using Graphical Displays of Data," *Proceedings of the 1996 Winter Simulation Conference*: 345-351 (1996a).

Sargent, Robert G. "Verifying and Validating Simulation Models," *Proceedings of the 1996 Winter Simulation Conference*: 55-64 (1996b).

Scott, E. Marian. "Uncertainty and Sensitivity Studies of Models of Environmental Systems," *Proceedings of the 1996 Winter Simulation Conference*: 255-259 (1996).

Shapiro, Alexander. "Simulation Based Optimization," *Proceedings of the 1996 Winter Simulation Conference*: 332-336 (1996).

Smart, D. R. *Fixed Point Theorems*. Cambridge: Cambridge University Press, 1974.

Taylor, Michael K., Paul F. Auclair, and Edward F. Mykytka. "Working Smarter when Developing Linear Simulation Metamodels," *Proceedings of the 1995 Winter Simulation Conference*: 1392-1398 (1995).

Van Groenendaal, Wilhelm J. H., and Jack P. C. Kleijnen. "Regression Metamodels and Design of Experiments," *Proceedings of the 1996 Winter Simulation Conference*: 1433-1439 (1996).

Vita

Major Samuel A. Wright [redacted] graduated from Whitehall-Yearling High School in Whitehall, Ohio in 1985 and attended the United States Air Force Academy, graduating with a Bachelor of Science in Astronautical Engineering in May 1989. Upon graduation, he received a regular commission in the United States Air Force. Major Wright's first tour of duty was with Air Force Space Command's 1st Space Operations Squadron at Falcon AFB. Major Wright held duties as a Planner/Analyst and Mission Commander for Global Positioning System (GPS) satellite operations. He also served as Chief, GPS Standardization/ Evaluation Branch, and Deputy Chief of the Engineering Branch for GPS satellites. Major Wright attended Squadron Officers' School before entering the Air Force Institute of Technology (AFIT) School of Engineering in August 1993. He graduated from AFIT in March 1995 with a Master of Science in Operations Research, at which time he entered the Department of Operational Sciences Ph.D. program. In July 1998, Major Wright was assigned to Scott AFB to serve the Air Mobility Command Studies and Analysis Flight as a Mobility Analyst, then as Senior Analyst for Future Air Mobility Operations.

[redacted]

[redacted]

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*
20-03-2001

2. REPORT TYPE
Doctoral Dissertation

3. DATES COVERED *(From – To)*
April 1996 – December 2000

4. TITLE AND SUBTITLE

COVALIDATION OF DISSIMILARLY STRUCTURED MODELS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Wright, Samuel A., Major, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/DS/ENS/00-02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

HQ AMC/XPY
402 Scott Drive, Unit 3L3
Scott AFB IL 62225
(618) 229-4319
amc-xpy@scott.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A methodology is presented which allows comparison between models constructed under different modeling paradigms. Consider the following situation: Two models are constructed to study different aspects of the same system. One model simulates a fleet of aircraft moving a given combination of cargo and passengers from an onload point to an offload point. A second model is a linear programming model that optimizes the aircraft and route selection required for the same scenario. We develop a methodology to structure the comparison between large-scale models such as these.

Models that compare favorably using this methodology are deemed covalid. Models that perform similarly under the same input conditions are covalid in a narrow sense. Models that are covalid (in this narrow sense) hold the potential to be used in an iterative fashion to improve the input (and thus, the output) of one another. We prove that, under certain regularity conditions, this method of output/input crossflow converges, and if the convergence is to a valid representation of the real-world system, the models are covalid in a wide sense. Further, if one of the models has been independently validated, then we may effect a validation by association of the other model through this process.

15. SUBJECT TERMS
Models, Model Theory, Model Comparisons, Simulation, Optimization, Validation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Dr. Kenneth W. Bauer, Jr., ENS, kenneth.bauer@afit.af.mil
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	154	(937) 255-6565, ext. 4328

	Form Approved *OMB No. 074-0188*